TRIBHUVAN UNIVERSITY

INSTITUE OF ENGINEERING

CENTRAL CAMPUS, PULCHOWK

MUSIC CLASSIFICATION BASED ON GENRE AND MOOD

Submitted By:

Ayush Shakya	[069/BCT/505]
Bijay Gurung	[069/BCT/512]
Mahendra Singh Thapa	[069/BCT/519]
Mehang Rai	[069/BCT/524]

A PROJECT WAS SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULLFILLMENT OF THE REQUIREMENT FOR THE BACHELORS DEGREE IN COMPUTER ENGINEERING

(August, 2016)

TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS
DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

The undersigned certify that they have read, and recommended to the Institute of Engineering for acceptance, a project report entitled ”Music Classification Based on Genre and Mood” submitted by Ayush Shakya, Bijay Gurung, Mahendra Singh Thapa and Mehang Rai in partial fulfilment of the requirements for the Bachelors degree in Computer Engineering.

Supervisor, Dr. Basanta Joshi
Department of Electronics and Computer Engineering

Internal Examiner
Name of Internal Examiner
Title, Affiliation

External Examiner
Name of External Examiner
Title, Affiliation

Dr. Nanda Bikram Adhikari
Deputy Head
Department of Electronics & Computer
Engineering,
Institute of Engineering, Pulchowk Campus,
Tribhuvan University, Nepal

Dr. Dibakar Raj Pant
Head
Deparment of Electronics & Computer
Engineering,
Institute of Engineering,Pulchowk Campus,
Tribhuvan University, Nepal

Date of Approval:

COPYRIGHTS

The author has agreed that the Library, Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering may make this report freely available for inspection. Moreover, the author has agreed that permission for extensive copying of this project report for scholarly purpose may be granted by the supervisors who supervised the project work recorded herein or, in their absence, by the Head of the Department wherein the project report was done. It is understood that the recognition will be given to the author of this report and to the Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering in any use of the material of this project report. Copying or publication or the other use of this report for financial gain without approval of to the Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering and authors written permission is prohibited.

Request for permission to copy or to make any other use of the material in this report in whole or in part should be addressed to:

Head
Department of Electronics and Computer Engineering
Pulchowk Campus, Institute of Engineering
Lalitpur, Kathmandu
Nepal

ACKNOWLEDGMENT

It has been a great pleasure to work with different individuals whose perpective and ideas has directly or indirectly assisted or motivated us. We would like to gratitude to everyone whose ideas lead to this completion of the project. It is a matter of fact a great privilege for us to acknowledge their assistance and contributions to our project.

First and foremost, we would like to express our sincere gratitude towards Dr. Basanta Joshi, our supervisor under whose supervision there was successful completion of our project. Without his invaluable guidance and suggestions, it would have been a difficult journey for us. His useful suggestions for this whole work and cooperative behavior are sincerely acknowledged.

We would like to thank the Department of Electronics and Computer Engineering for adding this major project as part of final year curriculum and hence giving us this opportunity to undertake the project. The great need of research, time and sheer coding has allowed us to harness our skills, experience and knowledge.

We are also grateful to Dr. Nanda Bikram Adhikari for letting us carry out this project and co-operating with us to carry out the project smoothly. We would like to thank and express our gratitude to all out respective subject teachers for sharing their precious knowledge, constant support and guidance.

Last but not the least, we would like to thank our friends for motivating us and providing numerous assistance throughout the project development duration.

ABSTRACT

This report describes and documents all the aspects and working functionality of our final year project titled Music Classification System Based on Genre and Mood. The project is part for the curriculum for the subject Major Project under the course of final year of B.E. in Computer Engineering. As the title itself describes the overall aim of the project is to develop a system capable of classifying music based on Genre and mood, with the availability of large number of digital media and the disorder introduced being the primary motivation.

The methodology used is that of a modular system consisting of two main stages. The first stage involves the preprocessing of the raw audio data resulting in the extraction of a number of features pertaining to music signal: Intensity, MFCC, rhythm, pitch. Each feature extractor reduces the information content in the raw data to a vector in a small number of dimensions. Or in other words we can say that feature extractor analyses the music signal and extracts its respective features compatible for further processing. It requires intensive knowledge of digital signal analysis and processing, signal sampling,etc. The second stage comprises of all the machine learning portion. In it, the set of feature vectors are classified(indexed) into certain clusters by the use of certain algorithms: K-means, Support Vector Machines and Artificial Neural Networks. This technically requires knowledge of all those respective algorithms.

This report also documents our approach towards the system development following the various aspects of Software Engineering. UML diagrams have been used to model the entire system and ERD diagrams have been used to show the relationship between the various entities in our system and iterative development method was chosen for the development of our system. Java language along with spring framework was used to build our whole system along with the GUI.

Contents

List of Figures

List of Tables

LIST OF ABBREVIATIONS

ADC	Analog to Digital Conversion
AMGC	Automatic Music Genre Classificaiton
ANN	Artificial neural network
API	Application Programming Interface
BPM	Beats Per Minute
DAG	Directed Acyclic Graph
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DSP	Digital Signal Processing
DWCH	Daubechies Wavelet Coefficient Histogram
EM	Expectation Maximization
FMA	Free Music Archive
FFT	Fast Fourier Transform
FT	Fourier Transform
GMM	Gaussian Mixture Model
GTZAN	George Tzanetakis
GUI	Graphic User Interface
ILR	Intermediate Level Requirement
ISMIR	International Symposium on Music Information Retrieval
JIT	Just In Time
JVM	Java Virtual Machine
KL	Kullback-Leibler
KNN	K-Nearest Neighbour
MER	Music Emotion Recognition
MFCC	Mel-frequency cepstral coefficients
MIDI	Musical Instrument Digital Interface

MIR	Music Information Retrieval
MPEG	Moving Picture Experts Group
MP3	MPEG Audio Layer 3
PCM	Pulse Code Modulation
RMS	Root Mean Square
SMILE	Statistical Machine Intelligence and Learning Engine
SONE	Specific Loudness sensation coefficients
STFT	Short Time Fourier Transform
SVM	Support vector machine
WAV	Waveform Audio File Format

LIST OF SYMBOLS

η	Learning rate
θ	Bias
Δ	Difference
δ	Artifical neural network node
τ	Lag at time period t

1 INTRODUCTION

1.1 Background

Music can be literally defined as the combination of soothing sounds. A more complex definition of music can be, a complex amalgam of melody, harmony, rhythm, timbre and silence in a particular structure. Music if an art form and cultural activity whose medium is sound and silence. It's a form of entertainment that puts sounds together in a way that people like or find interesting. To form a music, requirement of musical instruments are not necessary, for example a cappella, barbershop, choral, scat, plainsong, isicathamiya,etc. A group a person can simply sing in rhythm and form a music. Sometimes musician may use their voice to make noises similar to a musical intrument. Music gives the feeling of relaxation. For some people it momentarily stops the flow of time and for some it is a means of passage of time. We can find many music lovers all around the world. Its seems a bit abnormal if a person has no taste in music at all. Some might say a guitar is necessary to form a song or music, some might say a piano is a must, but some musicians may find music find in the chirping of bird, running water of river or even whistling of train.

The common elements of music are pitch (which governs melody and harmony), rhythm (and its associated concepts tempo, meter, and articulation), dynamics (loudness and softness), and the sonic qualities of timbre and texture (which are sometimes termed the ”color” of a musical sound). Different styles or types of music may emphasize, de-emphasize or omit some of these elements. Music is performed with a vast range of instruments and with vocal techniques ranging from singing to rapping, and there are solely intrumental pieces, solely vocal pieces. The creation, performance, significance, and even the definition of music vary according to culture and social context.

We may date the advent of music to be centuries year old. We can point it out due to the fact of presence of tribal music which has been passed on from generation to generation like the ancient African bushman tribal song, Nepalese traditional/cultural song from each race,etc. But however music can be complex. Though some may present a pattern in themselves like a chorus or say rift, some may have an uneven flow. Its perplex nature can not only due to its origination but also due to the evolution of music in different technological era. We have seen the evolution of music from legendary classical Beethoven symphonies to modern day hiphop which is widely popular among the youths nowadays. We have seen the rise of different genres like classical, rock, pop, metal,etc. and yet we may even don't know many of others at all and more may be yet too come like lately we have seen the rise of techno music.

It is not likely for a single person to listen to each and every genre present out there let alone all those songs. Every person may acquire a different taste in music. Some may like clasical music while come may like rock music, it's based on their choice. So a person may probably only distinguish a particular unknown song if those song were to belong to his/her genre of choice and the same goes for the mood. So realising this problem, there has been a increasing amount of research and work done in the music sector for the automatic classification of song based on genre. Though classification based on genre has been a popular one, classification based on mood catching the sight of many people lately.

With the advent of networks and internet, the number of songs are increasing exponentially throughout the internet. Website like soundcloud, youtube, facebook,etc. has given a platform for people to pursue their interest in music by forming like groups, composing and releasing songs, sharing songs,etc. Internet has made it possible for worldwide connection of whole world and it has harnessed the music industry. Because of internet only the popularity of music artists has been increasing all around the world. Their music have now been able to reach each and every corner of the world. This freedom of music throughout the internet has lead to increasing amount of songs and their databases. Due to these rapid development in the music industry, there has been an increasing amount of work in the area of automatic genre classification of music in audio format. A serious factor behind this automation can be considered as the increasing number of millions of records by different artist every year. A simple automation in classification would be much suited than a hand-to-hand task by human and its applicability can be huge. Moreover, there might be conflict regarding genre and mood issue based on the perception of a human being. So regarding these issues MIR(Music Information Retrieval) has primarily focused on automation of classification of such music based on signal analysis. Such systems can be used as a way to evaluate features describing music content as well as a way to structure large collections of music.

1.2 Overview

Throughout the evolution of music, the music industry took a different path and the difference in nature, flow of music, it's tempo, etc. is huge and quite complicated. It lead to the evolution of different numerous genre. The presence of numerous genre is a source of confusion and more often than not people are overwhelmed with the sheer vastness of music available. We humans can most of the times easily categorize simple song based on genre or mood by simply listening and analysing few sample of similar song based on similarity but we are never truly able to understand its nature or features distinction. So we can sometimes never able to recognize them correctly in case of genre and mood. There are songs out there for example Bohemian Rhapsody, which we can never really point it out to a distinct genre and mood.

Moreover the advent of internet has escalated the popularity of music and various artists. Nowadays everyone wants to be a singer. They want to become famous. So, various sites like soundcloud, youtube, facebook,etc. has provided them the perfect platform for sharing their songs. Not only for some novice singers but also for whole popular artist and whole music industry it has provided the perfect platform for sharing the music and growing itself. This has lead to release of millions of songs and increase in database of the system. So given the today world in computerized technological era, automation is nowadays seen as a popular subject in every field. There is being development of automation in every field like riding cars, manufacturing factories, etc. This popularity has affected the music industry too. Realising the potential of its applicability, there has been number of research in this sector/field. Numbers of research paper are being published regarding the automation of classification of music with research paper [29] published by George Tzanetakis and Perry Cook being one of the first in this field with the primary motivation to make it easier for people to classify music (based on genre and/or mood) so that they can find songs suited to their own tastes. It can also lay the foundation for figuring out ways to represent similarity between two musical pieces and in the making of a good recommendation system.

Given the perplexing nature of music, music classification requires specialized representations, abstraction and processing techniques for effective analysis, evaluation and classification that are fundamentally different form those used for other mediums and tasks. So focusing on these issues we created a music classification system which is web based application used for classifying music. We did not limit ourselves only to genre which is the burning issue in the music industry but we made our effort for the music classification based on mood too. In music industry there is a vast number of different genre. Most of the previous work were limited to four different genres. So, to challenge ourselves we took five different genres for our classification system, namely:-

- Classical
- Jazz
- Rock
- Pop
- Hiphop

Our application took a song as an input from the user computer and classified to its genre based on feature extracted and learning of the system.

Similar procedure was taken for classification of song based on mood. Until now not much research were done on music classification based on mood. So we made our classification system to classify that same song based on mood which was truly based on signal analysis and not lyrical features. For classification based on mood, we mapped the song among two dimensions:

- Energy
- Stress

So based upon the energy and stress level, our song is classified as:

- High Energy, High Stress = Anxious/Frantic

- High Energy, Low Stress = Exuberance
- Low Energy, High Stress = Depression
- Low Energy, Low Stress = Contentment

So, we can say that our system first extracted the required features based on the signal analysis and it's manipulation, and then used those features to classify it among one of the combination of five different genres and 4 different mood using the machine learning algorithm which is already trained on dataset.

1.3 Problem Statement

The evolution of music and its origination has presented us with many different genres. The advent and popularity of internet and networking has escalated the market and rise of music industry. Given the popularity of music industry, thousands of new artist are emerging every year. People are releasing song everytime as their hooby or part-time career. So we can see there are millions of songs out there world wide and is continuously increasing every year. Internet has huge contribution for it's rise. With that much of released songs, the size of database is also increasing every year. Since the subject of genre and mood depends on people's perception, it has really been a tedious job to create a quite standard one.

So we built a music classification system based on genre and mood. The choice of these genres is based on their being sufficiently distinguishable from each other. Choosing some genre thats very unique and abnormal might have made them more distinguishable and easier to classify but it would have been harder to find quality data/works for those genre. So, we chose these genre with availability of musical pieces in mind too. We chose to work on classification based on mood too because not many work had been done in the past regarding this field. But we can see this field has a wide scope of applicability. It can be used as a song recommendation system based on genre and that typical mood which the user is listening too as it is certain that the user will possibly like similar song with the similar melody. For now we are currently trying to tackle the issue of music classification based on genre and mood and not abiding to its applications.

1.4 Motivation

The presence of numerous number of different genres has presented tedious job for music industry. It has become a source of confusion and more often than not people are overwhelmed with the sheer vastness of music available. So, the primary motivation is to make it easier for people to classify music based on genre. Not only genre but classification based on mood has also now intrigued many people. Combined these two will provide or make a solid foundation for figuring out ways to represent similarity between two musical pieces and build a good recommendation system for music lovers who are passionate about their music and also their choices.

It can futher tackle the issue of automated music database management with large number. It can especially be useful in those cases with unknown label-genre and mood. Music player developers can then be able to make a smart playlist based on the genre and mood of some samples of song the user was currently or recently listening to. This would save a lot of time of user who had to otherwise manually maintain his/her playlist everytime based on his current mood and genre of choice.

1.5 Aims and Objectives

- To classify a music according to genre and mood.
- To create a web based application for music classification based on genre and mood.
- To extend the compatibility of the system with different types of music formats like wav,au,etc. along with mp3 format.

1.6 Scope of Project

(i) The project will work on classifying music based on genre and mood. More specifcally, the classifcation will be done on western music only as the data is more easily available and lots of works have been done in the past for it. Also, only five genres will be used for genre classifcation:

- Rock
- Pop
- Classical
- Jazz
- Hiphop

(ii) The mood based classifcation will use the Thayer model, a two dimensional model based on Energy and Stress:

- High Energy, High Stress = Anxious/Frantic
- High Energy, Low Stress = Exuberance
- Low Energy, High Stress = Depression
- Low Energy, Low Stress = Contentment

(iii) Also, it is entirely possible for a song or a piece of music to fall into multiple genre or moods. The characteristics that defne the genre and the mood may change within the song itself with one part showing seeming to belong to one class while other parts may seem to belong to an entirely diferent class. The project will not cover such issues. In other words, multiple-tagging will not be done.

(iv) The classification will work on various music file formats like mp3, au, wav,etc.

1.7 Organization of Report

This report describes and details the design and methodology of building a music classification system based on genre and method. As this report consists documentations relating to different field during development of a standard software product, hence the whole report is effectiely broke down to seven chapters.

Chapter one is intended to introduce the project by simply presenting a brief background of the project field which is music and music industry, the motivation which drove us to pursue the field, the overview of the problem statement and objective of the project and at last the scope of our project.

Chapter two presents the literature review. It provides us the collective effort that has been done in the past in our project field. Since our project is music classification based on genre and mood, so at first we start by brief history of human audio perception, Music Information Retrieval(MIR) and music classification. We give a general review of past activities and research on music classification based on both genre and mood. We describe the features and procedures have been involved in such system.

Chapter three is all about the system analysis done at perspective of software engineering. It describe about the requirement specification which is high level requirement, functional requirement and non functional requirement. It also involves feasibility assessment which contains operational feasibility, technical feasibility, economic feasibility, legal feasibility and scheduling feasibility.

Chapter four involves the methodology explaining the basic system block diagram and steps that are carried in the system like audio preprocessing, feature extraction and integration,etc.

Chapter five describes about the system development which all the use case diagram, data flow diagram and sequence diagram. It also explains about the tools and environment used for system development.

Chapter six involves the result and analysis process. Since our music classification is based on genre and mood, so we analyze the accuracy involved in each with each feature involved and also with different classifiers involved. We provide the details the all the testing and validation in it.
Finally, in chapter seven we present our conclusion. We present deduce a conclusion based on the result and analysis and give our insights on future enhancement of the system.

Along with all these there are list of references and bibliography relating to project which is included at last. There is also appendix provided which gives all the analysis and design diagrams which have been developed during the project.

2 LITERATURE REVIEW

2.1 Human Audio Perception

The human ear is an exceedingly complex organ. To make matters even more difficult, the information from two ears is combined in a perplexing neural network, the human brain. [1]

Figure 1 illustrates the major structures and processes that comprise the human ear. The outer ear is composed of two parts, the visible flap of skin and cartilage attached to the side of the head, and the ear canal, a tubeabout 0.5 cm in diameter extending about three cm into the head. These structures direct environmental sounds to the sensitive middle and inner ear organs located safely inside of the skull bones. Stretched across the end of the ear canal is a thin sheet of tissue called the tympanic membrane or eardrum. Sound waves striking the tympanic membrane cause it to vibrate. The middle ear is a set of small bones that transfer this vibration to the cochlea (inner ear) where it is converted to neural impulses. The cochlea is a liquid filled tube roughly two mm in diameter and three cm in length.

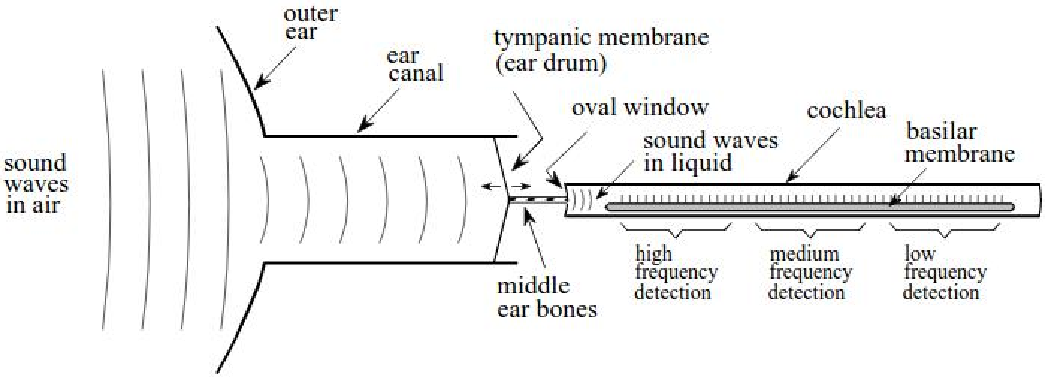


Figure 2.1: Functional Diagram of Human Ear

Music can be defined as organised sound comprising the following structural elements: pitch, timbre, key, harmony, loudness (or amplitude), rhythm, meter, and tempo. Processing these elements involves almost every region of the brain and nearly every neural subsystem.

Sound does not exist outside of the brain; it is simply air molecules moving. Sound is produced by vibrating air molecules connecting with the eardrum at varying frequencies (pitch) and velocities (amplitude). The process starts with the brains primary auditory cortex receiving a signal from the eardrum/inner ear which immediately activates our primitive brain, the cerebellum. The cerebellum is the oldest part of the brain in evolutionarily terms and plays an important part in motor control. It contributes to coordination, precision, and accurate timing of movements. The ear and the primitive brain are known collectively as the low-level processing units. They perform the main feature extraction which allows the brain to start analysing the sounds, breaking down the sensory stimulus into pitch, timbre, spatial location, amplitude, reverberant environment, tone durations, and onset times of different notes.

This data is conducted through neurons in the brain; cells specialized in transmitting information, and the basic building blocks of the nervous system. The output of these neurons connects to the high-level processing units located in the frontal lobe of the brain. It is important to note that this process is not linear. The different regions of the brain constantly update each other with new information.

2.2 History of MIR and Music Classification

The field of Music Information Retrieval (MIR) can be traced back to the 60s with reference to the works done by Kassler in [2]. Even Automatic Transcription of Music was attempted as early as the 70s [3]. However, there were two limiting factors that prevented progress in the field at the time. Firstly, the high computational requirements of the problem domain was simply not available. And secondly, other related fields of study such as Digital Signal Processing, Speech Processing, and Machine Learning were also not advanced enough. So, the field stalled for the next few decades.

In the 1990s, the field regained prominence as computational resources improved greatly and the rise of the internet resulted in massive online music collection. So, there was both an opportunity and demand for MIR systems. The organization of the first International Symposium on Music Information Retrieval (ISMIR 1) in 2000 highlights this resurgence of interest in the field. 280 people from 25 different countries participated in ISMIR Conference Malaga 2015.

As for the methodologies used, MIR in the 90s was influenced by the field of Text Information Retrieval (IR), techniques for searching and retrieving text documents based on user queries. So, most of the algorithms were developed based on symbolic representations such as MIDI files [4]. One such method is described in [5].

However, as mentioned in [6], identifying approximate units of meaning in MIR, as done by the majority of text-IR methods (words serve as such units) was not easy.

Instead, statistical non-transcriptive approaches for non-speech audio signals started being adopted in the second half on the 90s [4]. This was probably influenced by progress of such methods in other fields of speech processing. For example, in [7], the authors reported 98 per cent accuracy in distinguishing music from speech in commercial radio broadcasts. This was based on the statistics of the energy contour and the zero-crossing rate.

In [8], the authors introduced similar statistical methods for retrieval and classification of isolated sounds. Similarly, in [9], an algorithm for music-speech classification based on spectral feature was introduced. It was trained using supervised learning.

And so, starting in the 2000s, instead of methods attempting note-level transcriptions, researchers focused on direct extraction of information of audio signals using Signal Processing and Machine Learning techniques.

Currently, three basic strategies are being applied in MIR: [10]

- **Based on Conceptual Metadata** - Suited for low-specificity queries.
- **Using High-level Descriptions** - Suited for mid-specificity queries.
- **Using Low-level Signal-based Properties** - Used for all specificities.

But still most of the MIR techniques being employed at present use low-level signal features instead of high-level descriptors [11]. Thus, there exists a semantic gap between human perception of music and how MIR systems work.

2.3 Audio Processing

General Audio signal processing is an engineering field that focuses on the computational methods for intentionally altering sounds, methods that are used in many musical applications.

Particularly speaking, music signal processing may appear to be the junior relation of the large and mature field of speech signal processing, not least because many techniques and representations originally developed for speech have been applied to music, often with good results. However, music signals possess specific acoustic and structural characteristics that distinguish them from spoken language or other nonmusical signals. [20]

In music the most important qualities of sound are: pitch, duration, loudness, and timbre. Duration and loudness are unidimensional, while pitch and timbre are complex and multidimensional. [21]

- **Loudness** - Intensity of a tone is the physical correlate that underlies the perception of loudness. Loudness variations play an important role in music, but are less important than pitch variations.

- **Duration** - A composer or performer can alter the pace of a piece so that its apparent (virtual) time is slower or faster than clock time.

- **Timbre** - Timbre is the subjective code of the sound source or of its meaning. According to the American Standards Association, ”Timbre is that attribute of auditory senstation of which a listener can judge that two steady-state tones having the same pitch and loudness are dissimilar.”

- **Pitch** - Pitch is related to the frequency of a pure tone and to the fundamental frequency of a complex tone. In its musical sense, pitch has a range of about 20 to 5000 Hz. Some

> five to Zseven harmonics of a complex tone can be heard out individually by paying close attention. There is a dominance region for pitch perception, roughly from 500 to 2000 or 3000 Hz. Harmonics falling in the dominance region are most influential with regard to pitch.

Again, these types of low dimensional features extracted from the acoustical signals are more popular than higher dimensional representations such as Spectrograms for Classification purposes. [22]

2.4 Related Works

2.4.1 Genre Based Classification

2.4.1.1 Overview. Automatic Music Genre Classification (AMGC) is one of the tasks focused by MIR. However, it is not a straightforward one.

In [13], Scaringella et al. discuss how and why musical genres are a poorly defined concept making the task of automatic classification non-trivial. Still, although the boundaries between genres are fuzzy and there are no well-defined definitions, it is still one of the widely used method of classification of music. If we look at human capability in genre classification, Perrot et al [14] found that people classified songs–in a ten-way classification setup–with an accuracy of 70 per cent after listening to 3s excerpts.

2.4.1.2 Features. The features used for genre based classification have been heavily influenced by the related field of speech recognition. For instance, Mel-frequency Cepstral Coefficients (MFCC), a set of perceptually motivated features that is widely used in music classification, was first used in speech recognition.

The seminal paper on musical genre classification by Tzanetakis et al. [12] presented three feature sets for representing timbral texture, rhythmic content and pitch content. With the proposed feature set, they achieved a classification accuracy of 61 per cent for ten musical genre.

Timbral features are usually calculated for every short-time frame of sound based on the Short Time Fourier Transform (STFT). So, these are low-level features. Typical examples are Spectral Centroid, Spectral Rolloff, Spectral Flux, Energy, Zero Crossings, and the aforementioned Mel-Frequency Cepstral Coefficients (MFCCs). Among these, MFCC is the most widely preferred feature [15][16]. Logan [17] investigated the applicability of MFCCs to music modeling and found it to be ”at least not harmful”.

Rhythmic features capture the recurring pattern of tension and release in music while pitch is the perceived fundamental frequency of the sound. These are usually termed as mid-level features.

Apart from these, many non-standard features have been proposed in the literature.

Li et al.[18] proposed a new set of features based on Daubechies Wavelet Coefficient Histograms (DWCH), and also presented a comparative study with the features included in the MARSYAS framework. They showed that it significantly increased the accuracy of the classifier.

Anglade, Amlie, et al.[19] propose the use of Harmony as a high-level descriptor of music, focusing on the structure, progression, and relation of chords.

2.4.1.3 Classifer.

A variety of methods have been used for music classification. Some of the popular ones are SVM, K Nearest Neighbours and variants of Neural Networks. The results are also widely different. In [23], 61 per cent accuracy has been achieved using a Multilayer Perceptron based approach. While in [24], the authors have achieved 71 per cent accuracy through the use of an additional rejection and verification stage. Haggblade et al. [25], compared simpler and more naive approaches (k-NN and k-Means) with more sophisticated neural networks and SVMs. They found that the latter gave better results.

Standard statistical pattern recognition classifiers are also used for AMGC. They may be simple Gaussian Classifiers or Gaussian mixture model (GMM) classifier, where each class pdf is assumed to consist of a mixture of a specific number of multidimensional Gaussian distributions. In such an approach, the parameters of each Gaussian component and the mixture weights are estimated using the iterative EM algorithm.

However, lots of unique methods – either completely novel or a variation of a standard method – have been put into use too. In [26], the authors propose a method that uses Chord labeling (ones and zeros) in conjunction with a k-windowSubsequenceMatching algorithm used to find subsequence in music sequence and a Decision tree for the actual genre classification.

It is also noted that high-level and contextual concepts can be as important as low-level content descriptors. [19]

2.4.2 Mood Based Classification

2.4.2.1 Overview. As mood is a very human thing, Mood Based Classification, also known as Mood Emotion Recognition (MER), requires knowledge of both technical aspects as well as the human emotional system. So, the conceptualization of emotion and understanding of the associated emotion taxonomy is vital. However, it is a difficult thing to do, because

(i) It is subjective and

(ii) We cannot agree on a model to depict emotional states.

Usually, two approaces to emotion conceptualization are taken:

- **Categorical Conceptualization** - This approach to MER categorizes emotions into a number of distinct classes. It requires the belief of base emotions (happiness, anger, sadness, etc) from which all other secondary emotion classes can be derived.[30] However, the major drawback of the categorical approach is that the number of primary emotion classes is too small in comparison with the richness of music emotion perceived by humans.

- **Dimensional Conceptualization** - It defines Musical Values as numerical values over a number of emotion dimensions. So, the focus is on distinguishing emotions based on their position on a predefined space. Most of these conceptualizations map to three axes of emotions: valence (pleasantness), arousal (activation) and potency (dominance). By placing emotions on a continuum instead of trying to label them as discrete, this approach can encompass a wide variety of general emotions.

2.4.2.2 Circumplex and Thayer Mood Model.

One of the Dimensional conceptualization was proposed by Russell (1980) [31]. As shown in *Figure 2*, the model consists of a two-dimensional structure involving the dimensions of valence and arousal. General emotions are placed within thic circular framework.

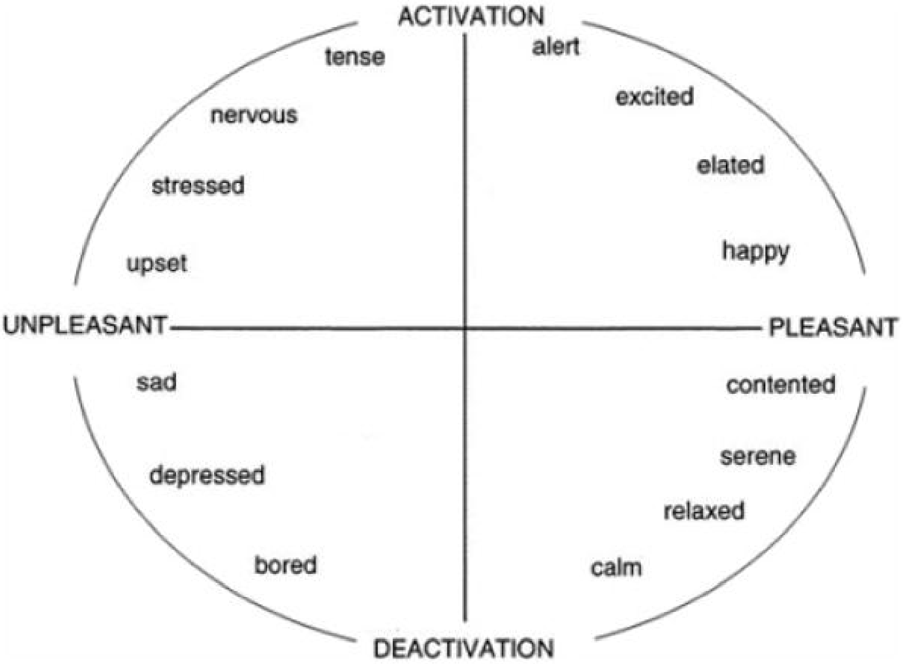


Figure 2.2: A graphical representation of the circumplex model of affect with the horizontal axis representing the valence dimension and the vertical axis representing the arousal or activation dimension.

As shown in *Figure 3*, Thayer [32] proposed a similar two-dimensional approach that adopts the theory that mood is entailed from two factors: -Stress (happy/anxious) -Energy (calm/ energetic). This divides music mood into four clusters: Contentment, Depression, Exuberance and Anxious/Frantic.

Although, the two-dimensional approach has been criticized as deficient (leading to a proposal of the third dimension of potency), it seems to offer the right balance between sufficient ”verbosity” and low complexity [33].

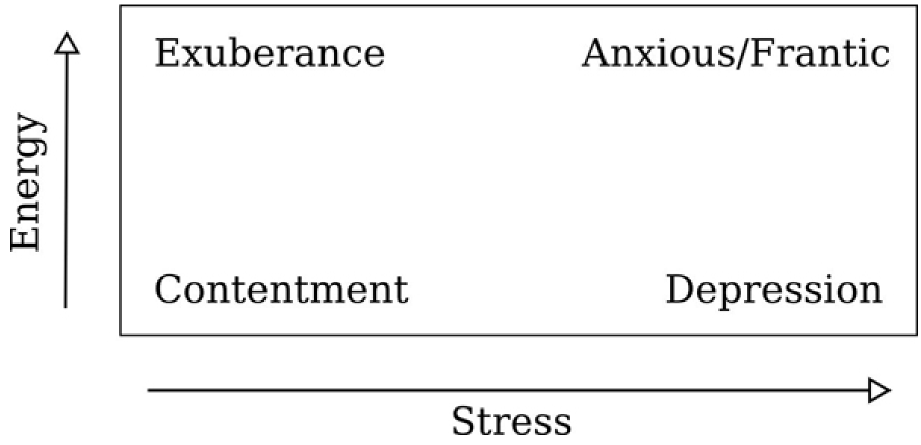


Figure 2.3: Thayer’s two-dimensional model of mood

2.4.2.3 Features.

Some of the commonly used features in MER are:

- **Energy**: Energy related features such as audio power, loudness, specific loudness sensation coefficients (SONE), are correlated to the perception of arousal. Lu et al. [34] used it to classify arousal.

- **Rhythm**: Flowing/fluent rhythm is associated with positive valence while firm rhythms with negative valence.

- **Melody**: These include features such as Pitch (perceived fundamental frequency), chromogram centroid, etc.

- **Timbre**: As with the AMGC problem, MFCC is widely used in MER too. Apart from MFCC, octave-based spectral contrast as well as DWCH (Daubechies wavelets coefficient histogram) are also proposed in literature.

So, we see that the features used in MER are almost the same as those in AMGC. However, Fu et al. note in their extensive survey on Audio-based Music Classification [35] that although their effectiveness is debatable, mid-level features such as Rhythm seem to be more popular in MER.

2.4.2.4 Classifiers. The algorithms used in AMGC are also popular in MER. So, support vector machines, Gaussian mixture models, neural networks, and k-nearest neighbor are the ones regularly used.

3 SYSTEM ANALYSIS

3.1 Requirement Specification

3.1.1 High Level Requirements

As our system is a music classification system, so we need our system to classify any given music file among the following genre and mood. Genres include:

- Hip-hop
- Rock
- Jazz
- Classical
- Pop

Mood include:

- Depressive
- Frantic
- Exuberant
- Contentment

The classification of the audio files based on genre and mood can be then be used in the creation of auto generated playlists.After each high level requirements are identified, corresponding intermediate level requirements (ILRs) are also identified. These list the feature metadata used to classify the audio.

3.1.2 Functional Requirements

- The system shall classify among the given genre and mood. Mood is classified in terms of arousal and valence.

- The system shall work on various music file format like mp3, wav, etc.

3.1.3 Non-Functional Requirements

Nonfunctional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.

The nonfunctional requirement in our project are:

- Code Documentation
- Project Documentation
- Code Quality
- Performance
- Fault Tolerance
- Scalability
- Testability
- Maintainability

3.2 Feasibility Assessment

A feasibility assessment or feasibility analysis is a preliminary study undertaken before the real work of a project starts to ascertain the likelihood fo the project's success. It is an analysis of possible alternative solutions to a problem and a recommendation on the best alternative. It, for example, can decide whether an order processing be carried out by a new system more efficiently than the previous one. It can be thought as an assessment of the practicality of a proposed project.

A feasibility study aims to objectively and rationally uncover the strengths and weakness of an existing project paradigm, opportunities and threats present in the environment, the resources required to carry through, and ultimately the prospects of success. In simplest terms, the two criteria to judge feasibility are cost required and value to be attained. A well-designed feasibility study should provide a historical background of the project, a description of the project and details of operations and technicality needed.Generally feasibility studies always precede technical development and project implementation. A feasibility study evaluates the project's potential for success. It must there be conducted with an objective, unbiased approach to provide information upon which decisions can be based.

The acronym TELOS refer to the five areas of feasibility:

- Technical feasibility
- Economic feasibility
- Legal feasibility
- Operational feasibility
- Scheduling feasibility

3.2.1 Technical Feasibility

After our thorough research we decided to go with the features mentioned above. We also chose simple but powerful algorithm to do the classification. Since our first priority is to do all the coding from the scratch, we do not guarantee to be optimised at highest level in terms of space and time complexity as we lack such expertise and experience of a professional developer. Since there is no standard procedure to check the accuracy of the extracted features, we might have hard time in validating the feature extraction process. But based on our experience we can do most of the coding by ourselves or at least try and then analyze the result. If we find any difficulty in future then we can use existing library like TarsoDSP,MP3SPI, etc. which are proved to provided excellent audio processing.

3.2.2 Economic feasibility

Being a research oriented project, our focus is soley based on knowledge and not on any monetary factors at all. Since our project is just the beginning portion which could be a huge leading factor in future and bring revolution in music industry, our current product might not be able to provide that much of montary benefit but sure it will be contributing to a huge research ahead in future. It's wide variety of application is sure to provide huge amount of benefits in future. Then, talking about the cost, as no hardware is required in our project so we can say there is almost no loss involved in monetary terms. Hence implementation can be started.

3.2.3 Operational Feasibility

Our system is not a hardware based. A computer is enough for it to run. We can say that there is no extra need for any hardware. Since we will be doing digital signal processing for the first we will be going to need the knowledge of signal processing.

3.2.4 Schedule feasibility

Our system is not that huge or complex. Therefore it will be easy to follow the pre-planned schedule for development. But however given our novice nature, we might lag a little behind the schedule provided a complex bug in our system arises.

3.2.5 Legal feasibility

Legal feasibility determines whether the proposed system conflicts with legal requirements. All of the algorithms does not have any copyright attached with them. We found that our conflicting factor might only be on the use of the database provided provided we misuse it or violate some conditions or if we illegally download it.

4 METHODOLOGY

4.1 System Block Diagram

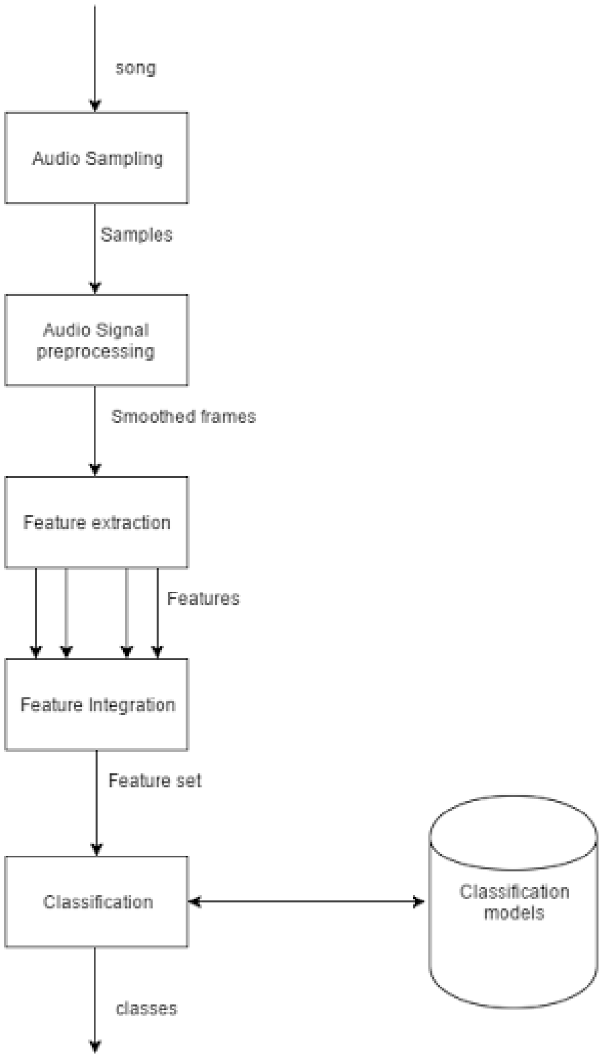


Figure 4.1: System block diagram

4.2 Audio Sampling

The central task that the Java sound API addresses is how to move bytes of formatted audio data into and out of the system. This task involves opening audio input and output devices and managing buffers that get filled with real-time sound data. It can also involve mixing multiple streams of audio into one stream (whether for input or output). The transport of sound into or out of the system has to be correctly handled when the user requests that the flow of sound be started, paused, resumed, or stopped.

To support this focus on basic audio input and output, the Java Sound API provides methods for converting between various audio data formats, and for reading and writing common types of sound files. However, it does not attempt to be a comprehensive sound-file toolkit. A particular implementation of the Java Sound API need not support an extensive set of file types or data format conversions. Third-party service providers can supply modules that ”plug in” to an existing implementation to support additional file types and conversions.

The Java Sound API can handle audio transport in both a streaming, buffered fashion and an in-memory, unbuffered fashion. ”Streaming” is used here in a general sense to refer to real-time handling of audio bytes.

Major components of the java sound API are:

(i) Formatted audio data refers to sound in any of a number of standard formats such as a data format which is represented by a Audioformat object which has the following attributes:

   - Encoding technique, usually pulse code modulation (PCM)
   - Number of channels (one for mono, two for stereo, etc.)
   - Sample rate (number of samples per second, per channel)
   - Number of bits per sample (per channel)
   - Frame rate
   - Frame size in bytes
   - Byte order (big-endian or little-endian)

   The file format on the other hand is represented using the Audio File Format which has the following attributes

   - The file type (WAVE, AIFF, etc.)

- The file's length in bytes
- The length, in frames, of the audio data contained in the file
- An AudioFormat object that specifies the data format of the audio data contained in the file

The Audio System class provides methods for reading and writing sounds in different file formats, and for converting between different data formats. Some of the methods let you access a file's contents through a kind of stream called an AudioInputStream .By reading a sound file as an AudioInputStream, you get immediate access to the samples, without having to worry about the sound file's structure (its header, chunks, etc.). A single method invocation gives you all the information you need about the data format and the file type.

(ii) Mixers which are used to handle one or more streams of audio input and one or more streams of audio output. In the typical case, it actually mixes together multiple incoming streams into one outgoing stream. A Mixer object can represent the sound-mixing capabilities of a physical device such as a sound card, which might need to mix the sound coming into the computer from various inputs, or the sound coming from application programs and going to outputs.

(iii) A line which is an element of the digital audio which is a path for moving audio into or out of the system. Usually the line is a path into or out of a mixer Audio input and output ports are lines. These are analogous to the microphones and speakers connected to a physical mixing console. Another kind of line is a data path through which an application program can get input audio from, or send output audio to, a mixer. These data paths are analogous to the tracks of the multitrack recorder connected to the physical mixing console.One difference between lines in the Java Sound API and those of a physical mixer is that the audio data flowing through a line in the Java Sound API can be mono or multichannel (for example, stereo). By contrast, each of a physical mixer's inputs and outputs is typically a single channel of sound. To get two or more channels of output from the physical mixer, two or more physical outputs are normally used (at least in the case of analog sound; a digital output jack is often multichannel). In the Java Sound API, the number of channels in a line is specified by the AudioFormat of the data that is currently flowing through the line.

Audio System serves as an application program's entry point for accessing these installed sampled-audio resources. You can query the Audio System to learn what sorts of resources have been installed, and then you can obtain access to them. For example, an application program might start out by asking the Audio System class whether there is a mixer that has a certain configuration, such as one of the input or output configurations illustrated earlier in the discussion of lines. From the mixer, the program would then obtain data lines, and so on.

Here are some of the resources an application program can obtain from the Audio System:

- **Mixers:** A system typically has multiple mixers installed. There is usually at least one for audio input and one for audio output. There might also be mixers that don't have I/O ports but instead accept audio from an application program and deliver the mixed audio back to the program. The Audio System class provides a list of all of the installed mixers.
- **Lines:** Even though every line is associated with a mixer, an application program can get a line directly from the Audio System, without dealing explicitly with mixers.
- **Format conversions:** An application program can use format conversions to translate audio data from one format to another.
- **Files and streams:** The Audio System class provides methods for translating between audio files and audio streams. It can also report the file format of a sound file and can write files in different formats.

4.3 Audio Signal Pre-processing

The preprocessing in music classification systems is used in order to increase the efficiency of subsequent feature extraction and classification stages and therefore to improve the overall system performance. Commonly the preprocessing includes the framing and window of the input signal. At the end of the preprocessing the smoothed frame are forwarded to the feature extraction stage.

The general preprocessing pipeline is depicted in the following figure.

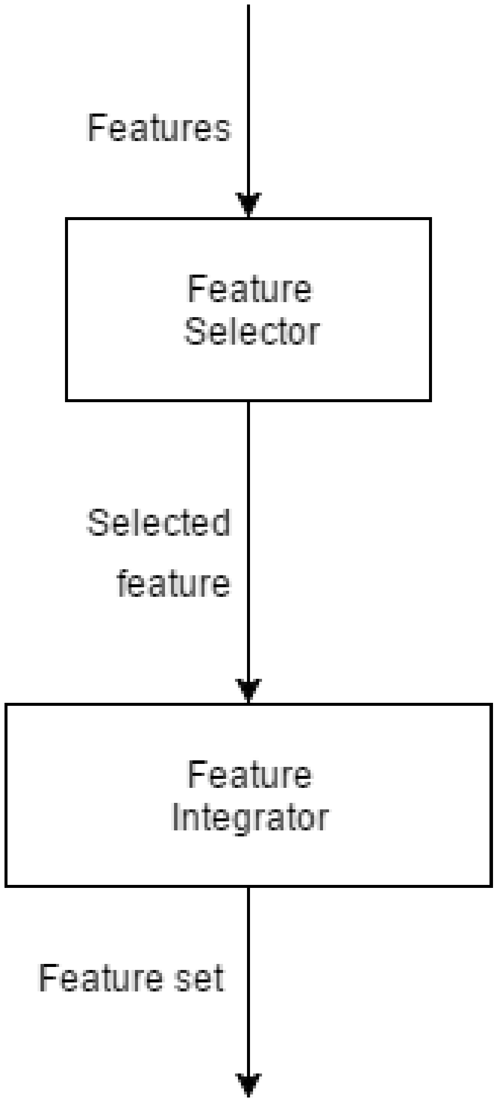


Figure 4.2: Feature integration

4.3.1 Framing

Framing is the process of dividing the whole audio sample into frames. An audio signal is constantly changing, so we assume that on short time scales the audio signal doesnt change (when we say it doesnt change, we mean statistically stationary, obviously the samples are constantly changing on even short time scales). This is why we frame the signal into 20-40 ms frames. If the frame is much shorter we dont have enough samples to get a reliable spectral estimate, if it is longer the signal changes too much throughout the frame.So, a frame provides a redundant, stable way of representing a signal. To mitigate the loss at the edges of the window, the frames may overlap in time.

4.3.2 Windowing

Windowing refers to the processing of multiplying of frames with the window function like rectangular window, hamming window, hanning window etc. Windowing are necessary because whenever we do a finite Fourier transform, it is implicitly being applied to an infinite repeating signal. So, for instance if the start and end of a finite sample doesnt match then that will look just like a discontinuity in the signal, and show up a lots of high-frequency noise in the Fourier Transform, which is harmful. If the sample happens to be a perfect sinusoid but with an integer number of periods then it doesnt fit exactly into the finite sample and the FT will show appreciable energy in all sorts of places nowhere near the real frequency. So windowing make sure that the end matchup while keeping everything reasonably smooth. where s(n) denotes the sampled speech signal,

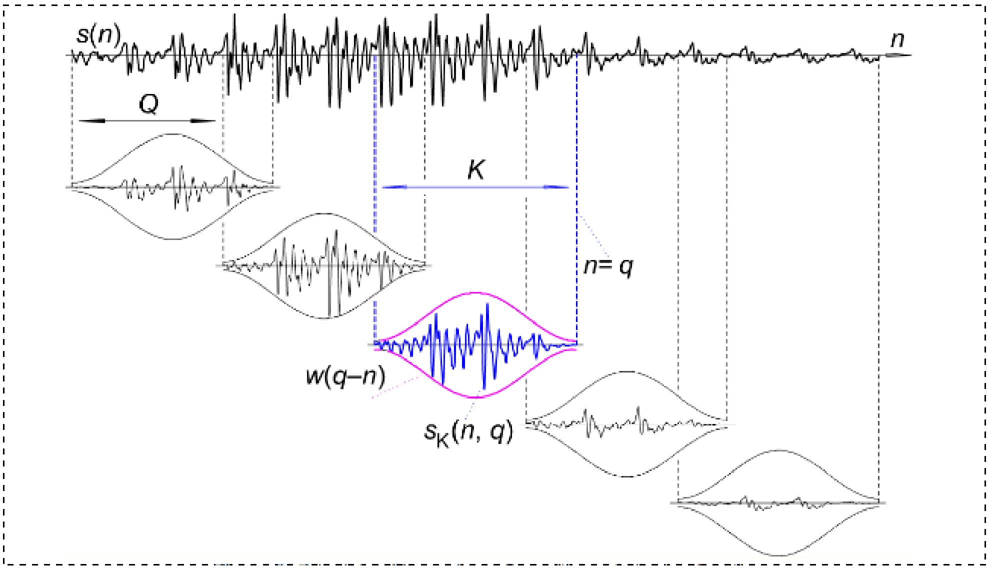


Figure 4.3: Principle of frame formation and windowing function

Q is the frame length,

K is the window length,

q is the sample point the window is applied,

and s_k(n,q) is a resulting short time signal, with s_k(n,q) = s(n)w(q-n).

4.4 Feature Extraction

Feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. Feature extraction is related to dimensionality reduction.

The feature extraction procedure is depicted in the following figure.

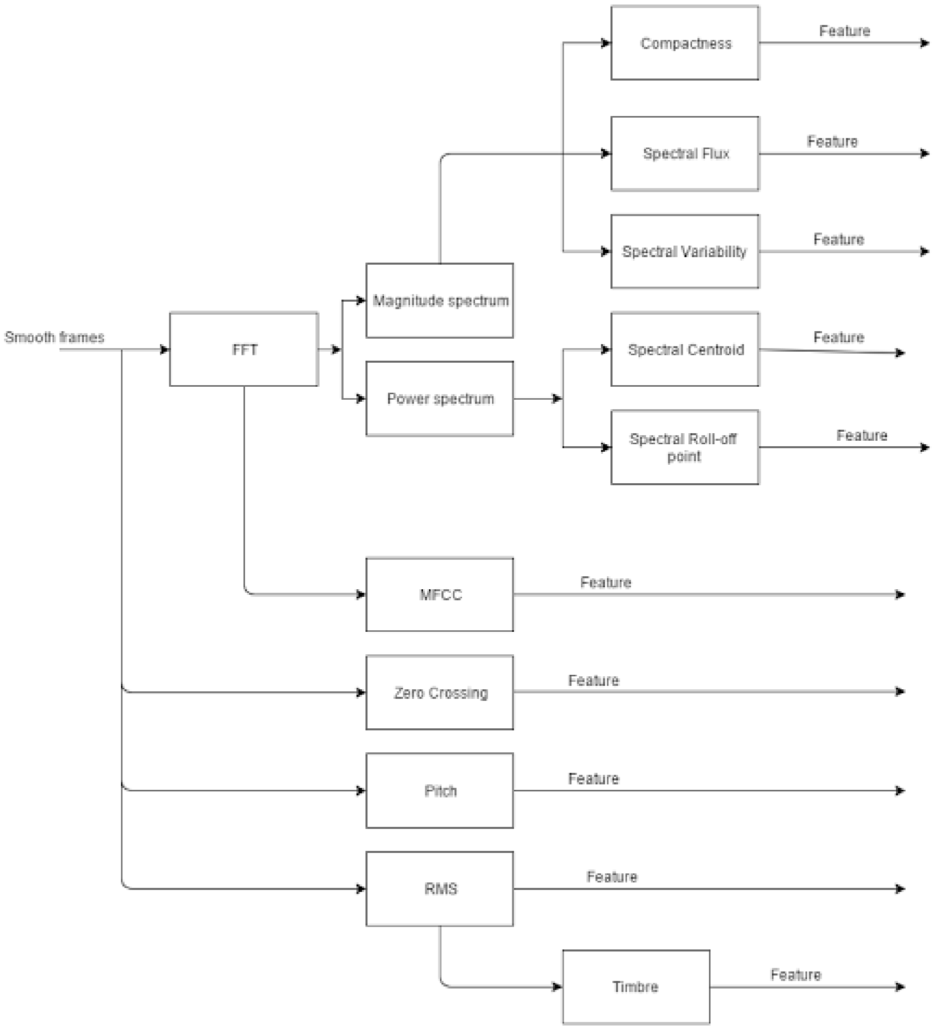


Figure 4.4: Feature extraction

4.4.1 Fast Fourier Transform

A Fast Fourier Transform(FFT) algorithm computes the DFT(Discrete Fourier Transform) of a sequence, or its inverse. Fourier analysis converts a signal from its original domain (often time or space) to a representation in the frequency domain and vice versa. An FFT rapidly computes such transformation by factorizing the DFT matrix into a product of sparse (mostly zero) factors. As a result, it manages to reduce the complexity of computing the DFT from O(n^2) which arises if one simply applies the definition of the DFT to O(nlogn), where n is the data size.

4.4.2 Magnitude Spectrum

Magnitude Spectrum is the magnitude of different frequency components within a window. The magnitude spectrum is found by first calculating the FFT within window. The magnitude spectrum value for each bin is found by first summing the squares of the real and imaginary components. The square root of this is then found and the result is divided by the number of bins.

4.4.3 Power Spectrum

Power Spectrum is the power of different frequency components within a window. The power spectrum is found by first calculating the FFT with a Hanning window. The magnitude spectrum value for each bin is found by first summing the squares of the real and imaginary components. The result is divided by the number of bins.

4.4.4 Compactness

It is the measure of the noisiness of a signal. It is found by comparing the components of a windows magnitude spectrum with the magnitude spectrum of its neighbouring windows.

If M[n], M[n-1] and M[n+1] > 0, then

$$compactness = \sum_{n=2}^{N-1} ((|20 * log(M[n])) - 20 * (log(M[n-1]) + log(M[n]) + log(M[]n+1))/3|) \tag{1}$$

otherwise,

$$compactness = 0 \tag{2}$$

where M[n] is the Magnitude Spectrum at internal n.

4.4.5 Mel-Frequency Cepstral Coefficients

It is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency.

Algorithm

(i) **Framing:** The process of segmenting the speech samples obtained from analog to digital conversion(ADC) into a small frame with the length within the range of 20 to 40 msec. The voice signal is divided into frames of Nsamples. Adjacent frames are being separated by M(M < N).

(ii) **Hamming Window:** Hamming window is used as window shape by considering the next block in feature extraction processing chain and integrates all the closest frequency lines. The Hamming window equation is given as:

If the window is defines as W(n), 0 ≤ n ≤ N-1 where
N = number of samples in each frame
Y[n] = Output signal X(n) = Input signal W(n) = Hamming window,
then the result of windowing signal is shown below:

$$Y(n) = X(n) \times W(n) \tag{3}$$

$$W(n) = 0.54 - 0.46cos\left(\frac{2\pi n}{N-1}\right), \qquad 0 \leq n \leq N-1 \tag{4}$$

(iii) **Fast Fourier Transform:** To convert each frame of N samples from time domain into frequency domain. The Fourier Transform is to convert the convolution of the glottal pulse U[n] and the vocal tract impulse response H[n] in the time domain. This statement supports the equation below:

$$Y(w) = FFT[h(t) * X(t)] = H(w) * X(w) \tag{5}$$

If X(w), H(w) and Y(w) are the Fourier Transform of X(t), H(t) and Y(t) respectively.

(iv) **Mel Filter Bank Processing:** The frequencies range in FFT spectrum is very wide and voice signal does not follow the linear scale as shown in figure 4.5 is then performed. This figure shows a set of triangular filters that are used to compute a weights sum of filter spectral components so that the output of process approximates to a Mel scale. Each filters magnitude frequency response is triangular in shape and equal to unity at the centre frequency and decrease linearly to zero at centre frequency of two adjacent filters [7,8]. Then, each filter output is the sum of its filtered spectral components. After that the following equation is used to compute the Mel for given frequency f in Hz: This figure shows a set of triangular filters that are used to compute a weights sum of filter spectral components so that the output of process approximates to a Mel scale. Each filter's magnitude frequency response is triangular in shape and equal to unity at the centre frequency and decrease linearly to zero at centre frequency of two adjacent filters [7,8]. Then, each filter output is the sum of its filtered spectral components. After that the following equation is used to compute the Mel for given frequency f in Hz:

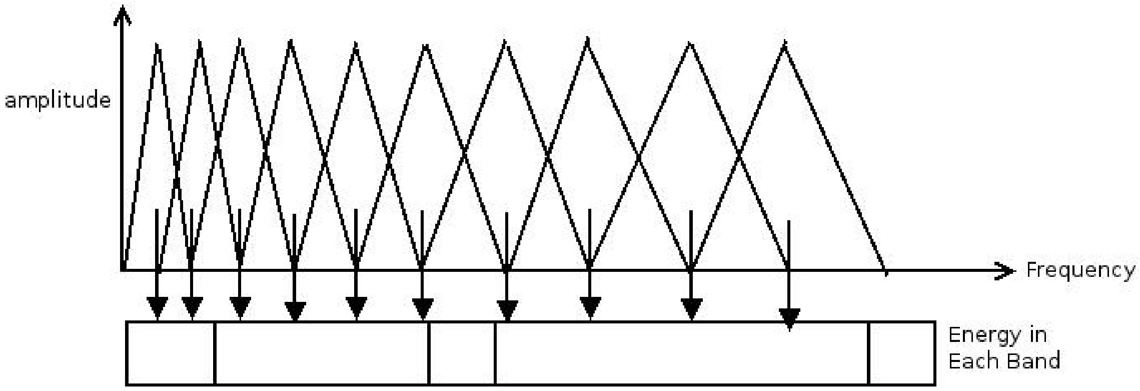


Figure 4.5: Mel scale filter bank

$$M(f) = 1125ln\left(1 + \frac{f}{700}\right) \tag{6}$$

(v) **Discrete Cosine Transform:** This is the process to convert the log Mel spectrum into time domain using Discrete Cosine Transform(DCT). The result of the conversion is called Mel Frequency Cepstrum Coefficient. The set of coefficient is called acoustic vectors. Therefore, each input utterance is transformed into a sequence of acoustic vector.

4.4.6 Pitch

It is a perceptual property of sounds that allows their ordering on a frequencyrelated scale, or more commonly, pitch is the quality that makes it possible to judge sounds as ”higher” and ”lower” in the sense associated with musical melodies.

It is a subjective psychoacoustical attribute of sound, and hence is approximately quantified as fundamental frequency.Pitch is an auditory sensation in which a listener assigns musical tones to relative positions on a musical scale based primarily on their perception of the frequency of vibration.Pitch is closely related to frequency, but the two are not equivalent. Frequency is an objective, scientific attribute that can be measured. Pitch is each person’s subjective perception of a sound wave, which cannot be directly measured. However, this does not necessarily mean that most people won’t agree on which notes are higher and lower.

Algorithm

(i) Model the signal x_t as a periodic function with period T, by definition invariant for a time shift of T

$$x_t - x_{t+T} = 0, \qquad \forall t \tag{7}$$

The same is true after taking the square and averaging over a window

$$\sum_{d=t+1}^{t+W} (x(j) - x(j+\tau))^2 = 0 \tag{8}$$

Conversely, an unknown period may be found by forming a difference function and searching for the values of τ for which the function is zero.

(ii) The cumulative mean normalized difference function is calculated by dividing each value of the old by its average over shorter-lag values. It differs from difference function in the first step in that it starts at one rather than zero, tends to remain large at low lags, and drops below one only where the first difference function falls below average.

(iii) Set an absolute threshold and choose the smallest value of τ that gives a minimum of the difference function obtained in the second step deeper than that threshold. If none is found, the global minimum is chosen instead. With a threshold of 0.1, the error rate drops significantly.

(iv) Each local minimum of the second difference function and its immediate neighbors is fit by a parabola, and the ordinate of the interpolated minimum is used in the dip-selection process. The abscissa of the selected minimum then serves as a period estimate. Actually, one finds that the estimate obtained in this way is slightly biased. To avoid this bias, the abscissa of the corresponding minimum of the raw difference function(the first) is used instead.

4.4.7 Tempo

The beat is the regularly occurring pattern of rhythmic stresses in music. When we count, tap or clap along with music we are experiencing the beat. Try tapping your finger along with different types of music and see what happens.

Tempo is the speed of the beat, usually expressed in Beats Per Minute(BPM). For example, at 120 BPM there will be 120 beats in one minute. Tempo can also be expressed verbally with different music terms, such as Slowly, Fast, Allegro, or Largo.

Algorithm

(i) Parse an audio file into samples, s[n], with a corresponding sampling rate, SR.

(ii) Break the audio file into N windows of 1024 samples.

(iii) Calculate the FFT of each window.

(iv) Calculate the power spectrum(P) of each window form the corresponding FFT results.

(v) Calculate the spectral flux(F) from the power spectrum(P) of each window, i:

$$F_i = (P_i - P_{i-1})^2 \tag{9}$$

(vi) Find the mean flux, F_{av}, across all windows.

(vii) Set F_i, the flux for each window, to 0 if it is not at least 1.5 times F_{av} (this value was experimentally determined). This gives a generous estimation of note onsets.

(viii) Use autocorrelation to find the histogram of lag frequencies, L:

$$L[lag] = \sum_{i}^{N} F[i]F[i+lag] \tag{10}$$

(ix) Calculate the effective sampling rate, S_{eff}, found in F and L.

$$SR_{eff} = \frac{SR}{1024} \tag{11}$$

(x) Convert the lag histogram L[lag] into a tempo histogram L[BPM] with bins of beats per minute by reversing the order of L and converting the bin lag indices, lag, to BPM:

$$L[BPM] = L\Big(\frac{60 * SR_{eff}}{lag}\Big) \tag{12}$$

(xi) The result of step (x), L[BPM], is a tempo histogram with bin labels corresponding to beats per minute and bin frequencies correspoding to frequencies of inter-peak intervals.

4.4.8 Root Mean Square

The root mean square(abbreviated RMS of rms) is defined as the square root of mean square(the arithmetic mean of the squares of a set of numbers).

$$x_{rms} = \sqrt{\frac{1}{n}(x_1^2 + x_1^2 + ... + x_n^2)} \tag{13}$$

where there is set of n values $\{x_1, x_2, ..., x_n\}$.

4.4.9 Spectral Centroid

The spectral centroid is defined as the center of gravity of the magnitude spectrum of the STFT.

$$C_t = \frac{\sum_{n=1}^{N} M_t[n] * n}{\sum_{n=1}^{N} M_t[n]} \tag{14}$$

where $M_t[n]$ is the magnitude of the Fourier transform at frame t and frequency bin n.

The centroid is a measure of spectral shape and higher centroid values correspond to ”brighter” textures with more high frequencies.

4.4.10 Spectral Flux

The spectral flux is defined as the squared difference between the normalized magnitudes of successive spectral distributions.

$$F_t = \sum_{n=1}^{N} (N_t[n] - N_{t-1}[n])^2 \tag{15}$$

where $N_t[n]$ and $N_{t-1}[n]$ are the normalized magnitude of the Fourier transform at the current frame t, and the previous frame t-1, respectively.
The spectral flux is a measure of the amount of local spectral change.

4.4.11 Spectral Roll-off Point

The spectral roll-off is defined as the frequency R_t below which 85 per cent of the magnitude distribution is concentrated.

$$\sum_{n=1}^{R_t} M_t[n] = 0.85 * \sum_{n=1}^{N} M_t[n] \tag{16}$$

The roll-off is another measure of spectral shape.

4.4.12 Spectral Variability

Spectral variability is the standard deviation of the magnitude spectrum. This gives the measure of the variance of a signal's magnitude spectrum.

$$Spectralvariability = \sqrt{\frac{1}{N-1} - (\sum_{n=1}^{N} M[n] - mean)^2} \tag{17}$$

where M[n] = Magnitude spectrum of the signal at interval n, Mean = mean of the magnitude spectrum.

4.4.13 Zero Crossing

The zero crossing is defined as the number of times the waveform changed sign.

$$Z_t = \frac{1}{2} \sum_{n=1}^{N} |sign(x[n]) - sign(x[n-1])| \tag{18}$$

where the sign fucntion is 1 for positive arguments and 0 for negative arguments and x[n] is the time domain signal for frame t.

Time domain zero crossings provide a measure of the noisiness of the signal.

4.5 Feature Integration

Feature Integration is the processing of integrating the various significant features obtained by the feature extraction process.

The feature integration procedure is depicted in the following figure.

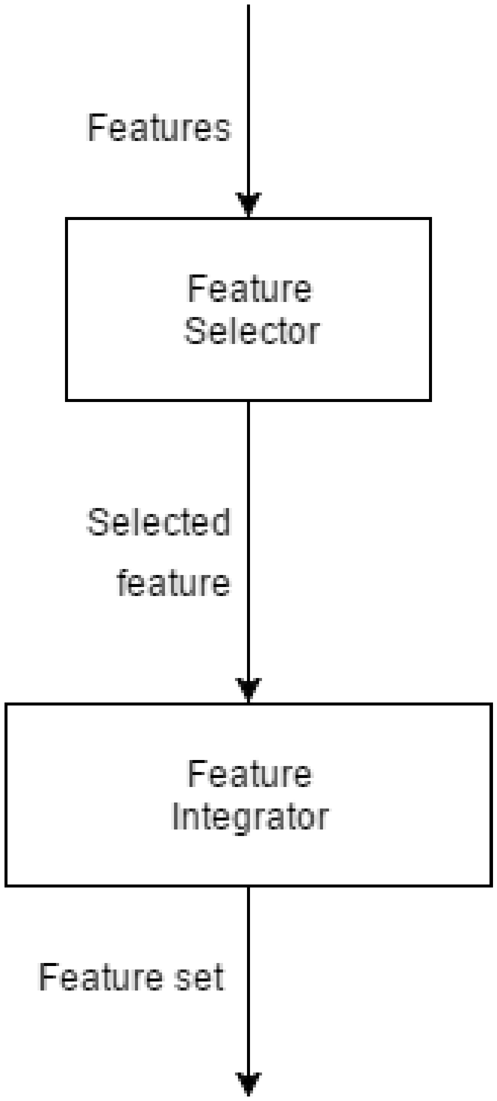


Figure 4.6: Feature integration

4.5.1 Feature Selector

Feature selector perform the feature selection process. Feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features(variable, predictors) for use in model construction. Feature selection technique are used for the three reasons:

(i) Simplification of models to make them easier to interpret by researchers/users

(ii) Shorter training time

(iii) Enhanced generalization by reducing overfitting

4.5.2 Feature Integrator

Feature integrator perform the feature integration process.Basically the features are temporal, so there is temporal feature integration. Feature integration is the process of combining all the feature vectors in a time window into a single feature vector in order to capture the relevant temporal information in the window. Here the mean and variance are used for temporal feature integration, but they capture neither the temporal dynamics nor dependencies among the individual feature dimensions.

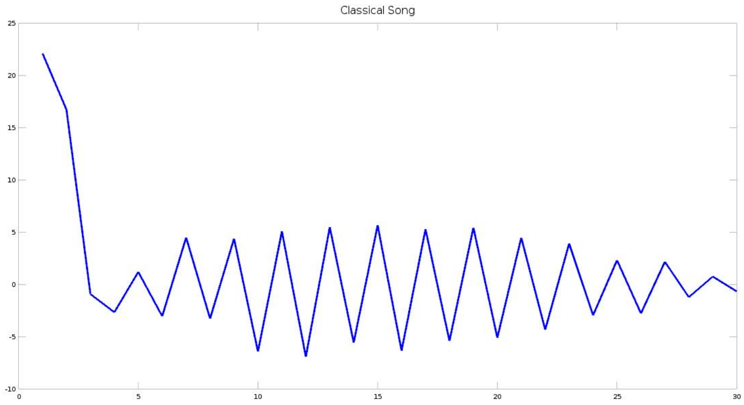


Figure 4.7: Mean of MFCC of classical

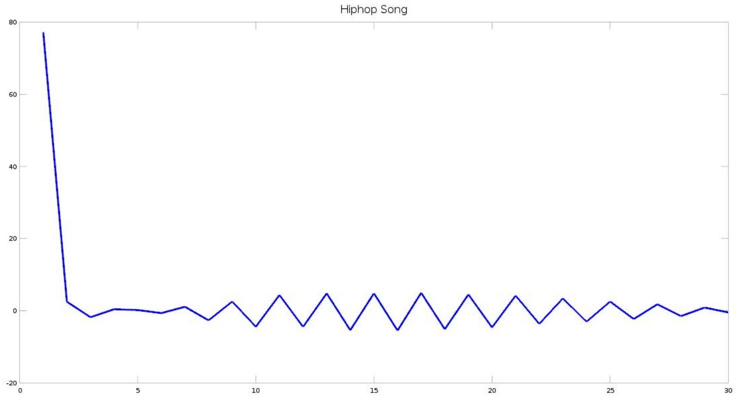


Figure 4.8: Mean of MFCC of hiphop

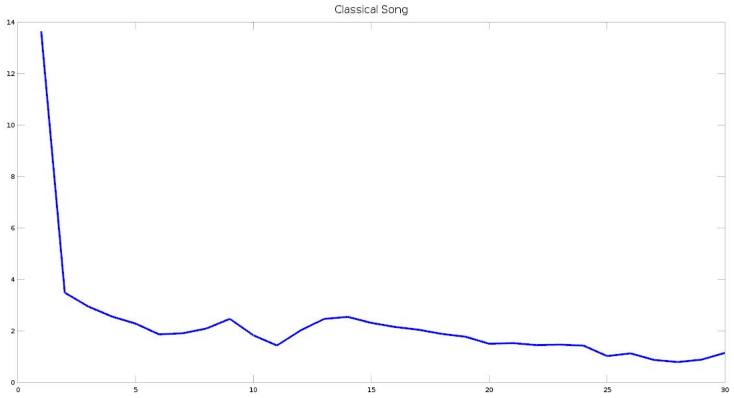


Figure 4.9: Standard deviation of MFCC of classical

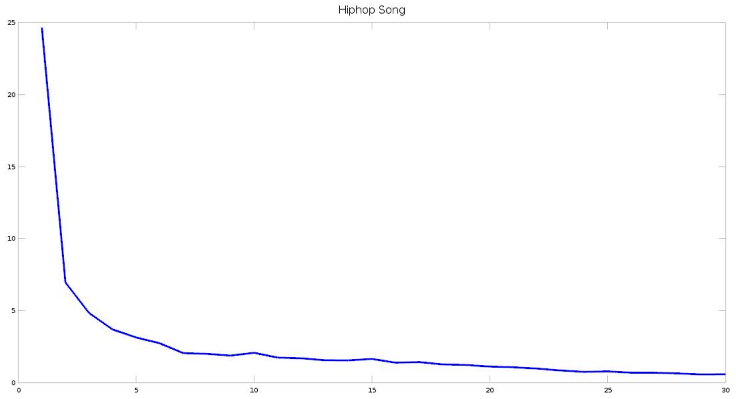


Figure 4.10: Standard deviation of MFCC of hiphop

4.6 Classification

4.6.1 ANN model

It stores the relevant parameters obtained from ANN training for future use without having to retrain.

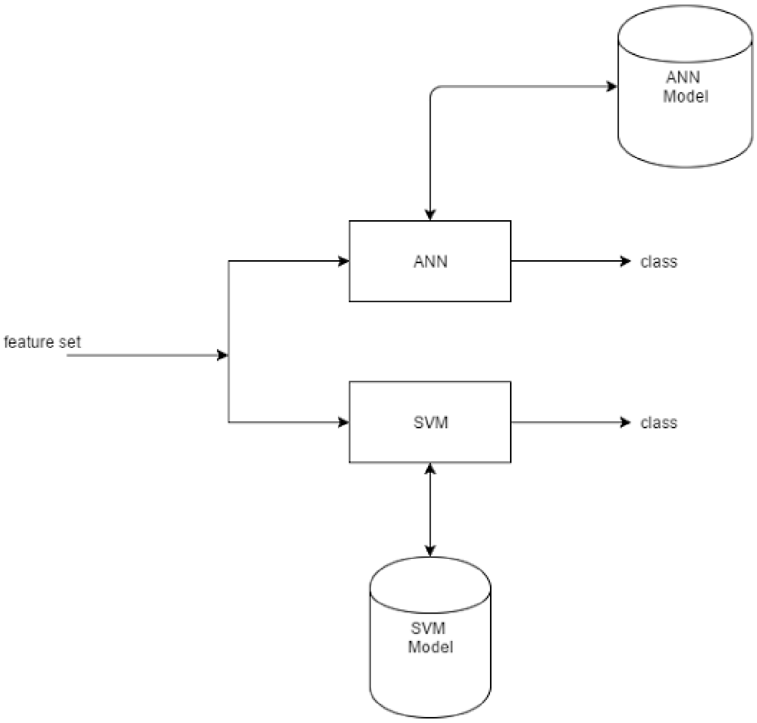


Figure 4.11: Classification

4.6.2 SVM model

It stores the relevant parameters obtained from SVM training for future use without having to retrain.

4.6.3 Artificial Neural Network.

In machine learning and cognitive science, an artificial neural networks(ANN) is a network inspired by biological neural networks(the central nervous systems of animals, in particular the brain) which are used to estimate or approxiamte functions that can depend on a large numbers of inputs that are generally unknown. Our research shows whether the research is related to our field or other, most of the time for machine learning researchers used artificial neural network. So our choice was simple as it was based on supervised learning and comparatively simple than other complicated unsupervised learning. Research papers [24], [23], ,[19], [25] and [16] shows the implementation of artificial neural network in the field of music classification.

Artificial neural networks are typically specified using three things:

- **Architecture:** It specifies what variables are involved in the network and their topological relationships- for example the variable involved in a neural network might be the weights of the connections between the neurons, along with activities of the neurons. In an artificial neural network, there are one or more hidden layers in between input and output layers with all the neurons connecting to each other.
- **Activity Rule:** Most neural network models have short tiem-scale dynamic: local rules define how the activities of the neurons change in reponse to each other. Typically the activity rule depends on the weights(the parameters) in the network.Here in our case, the set of input neuronse is activated by the feature values like MFCC, pitch,etc. There activation function present to trigger the respective neuron. Most of the time activation function are non linear, differential mathematical functions like sigmoid, hyperbolic tangent, etc.
- **Learning Rule:** The learning rule specifies the way in which the neural network's weights change with time as the learning takes progress. This learning is usually viewed as taking place on a longer time place than time scale of the dynamicx under the activity rule. Usually the learning rule will depend on the activites of the neurons. In our case the learning depends on the values of the target values supplied by a teacher and on the current value of the weights.

For a system, generally the architecture is created as per the need. Generally for that there is lot of trial and error methodology involved to determine the correct number of hidden layers and neurons needed. In most cases, one hidden layer is suffiecient for the system but if the nature of system or data is perplexing then in accordance to expected result and current behavior, number of hidden layers of nodes can be increased.

The concept of cost function is an important in context of artificial neural network. It measures how far away a particular solution is from an optimal solution. We can also say that the cost of the optimal solution is minimum. So, the target of our artifical neural network

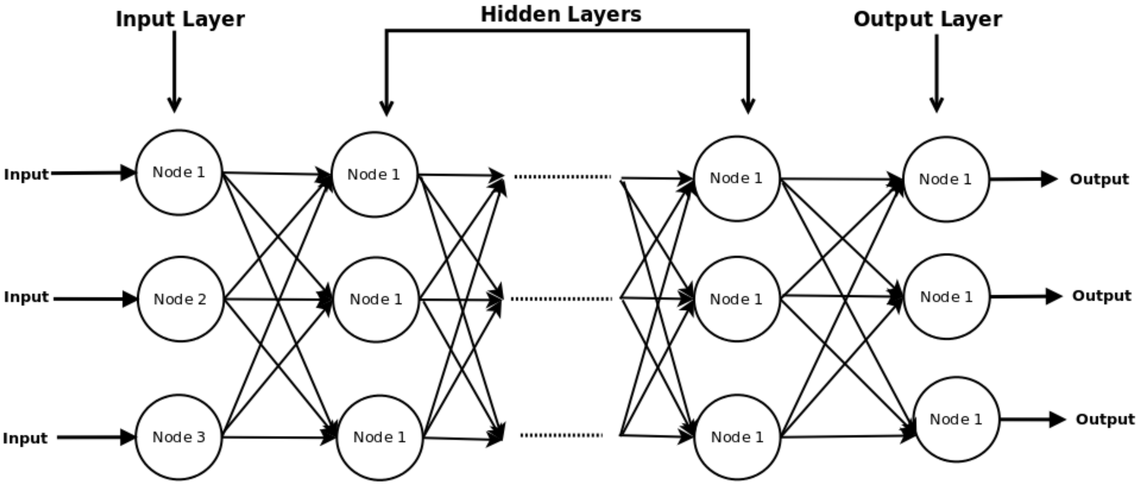


Figure 4.12: General structure of artificial neural network

is to try to meet the cost of optimal solution. While it is possible to define some arbitrary ad hoc cost fucntion, frequently a particular cost will be used, either because it has desirable properties (such as convexity) or because it arises naturally from a particular formulation of problem. Ultimately, the cost fucntion will depend on the desired task. One of the mostly used cost function is squared error measure between the output value O and the target value t.

$$E = (t - y)^2 \tag{19}$$

where E is the discrepancy or error.

Using this cost, the neural network tries to adjust it weights in order to minimize the cost function. This exact process is called learning. Supervised learning, unsupervised learning and reinforcement learning are the three major paradigms of learning. We are opting for the supervised learning. The reason for this choice is that unsupervised learning and reinforcement learning are comparatively more complex and also supervised learning have been doing great job in this music classification field based on research papers [23], [25], [16]. Supervised learning requires the correct class label to be provided along with the training dataset for the neural network so that it can adjust it's weight based on the cost function of incorrect/correct class prediction. Backpropagation algorithm is the most popular learning algorithm for neural network out there. The reason for its popularity might be its simplicity in terms of concept and wide applicability. It's also based on supervised learning. When one tries to minimize cost function using gradient descent for the class of neural networks called multilayer perceptrons(MLP), one obtains the common and well-known backpropagation algorithm for training neural networks.

Backpropagation is a common method of training artificial neural networks used in conjunction with an optimization method such as gradient descent. The method calculates the gradient of a loss function with respect to all the weights in the network. The gradient is fed to the optimization method wihich in turn uses it to update the weights, in an attempt to minimize the loss function. It is a generalization of delta rule to multi-layered feedforward networks, made possible by using the chain rule to iteratively compute gradients for each layer. Requirements of backpropagation mehod are:

- It requires a known, desired output for each input value in order to calculate the loss function gradient.
- It requires the activation function used by the artificial neurons/nodes to be differentiable.

Algorithm

(i) Run the network forward with your input data to get the network output.

(ii) For each output node compute

$$\delta_k = O_k(1-O_k)(O_k-t_k) \tag{20}$$

(iii) For each hidden node calculate

$$\delta_j = O_k(1-O_k)\sum_{k\in K}\delta_k W_{jk} \tag{21}$$

(iv) Update the weights and biases as follows:
Given

$$W = -\eta\,\delta_l O_{l-1} \tag{22}$$

$$\Delta\theta = -\eta\,\delta_l \tag{23}$$

apply

$$W + \Delta W \rightarrow W \tag{24}$$

$$\theta + \Delta\theta \rightarrow \theta \tag{25}$$

where i, j and k represents the input layer, hidden layer and ouput layer respectively,
O is the output of a neuron/node,
t is the target value,
K is the total number of output neurons/nodes,
Δ represent the difference,
l denotes every layer.

η is the learning rate which is defined as the ratio(percentage) that influences the speed and quality of learning.The greater the ratio, the faster the neuron trains; the lower the ratio, the more accurate the training is. θ is the bias term which is involved in adjusting the shifting of activation function. The sign of the gradient of a weight indicates where the error is increasing, this is why the weight must be updated in the opposite direction. The algorithm above is repeated until performance of the network is satisfactory.

4.6.4 Support Vector Machine.

In machine learning, support vector machines (SVM) are supervised learning models with associated learning algorithms that analyze data used for classification analysis.They are new statistical learning technique that can be seen as a new method for training classifiers based on polynomial functions, radial basis functions, neural networks, splines or other functions. The popularity of Support Vector machine is huge as lot of researcher papers [24], [19], [25], [16] shows its implementation. Not only in the field of music classification but also on various artificial intelligence field like handwriting recognition, biological and other sciences. It was also able to overthrow general neural network until the advent of deep learning.

The basic support vector machine (SVM) is a binary linear classifier which chooses the hyperplane that represents the largest separation, or margin, between the two classes. So Support vector machines use a hyperplane to create a classifier. If such a hyperplane exists, it is known as the maximummargin hyperplane and the linear classifier it defines is known as a maximum margin classifier. If there exists no hyperplane that can perfectly split the positive and negative instances, the soft margin method will choose a hyperplane that splits the instances as cleanly as possible, while still maximizing the distance to the nearest cleanly split instances. For problems that cannot be linearly separated in the input space, this machine offers a possibility to find a solution by making a non-linear transformation of the original input space into a high dimension feature space where an optimal separating hyper plane can be found. Those separating planes are optimal, which means that a maximal margin classifier with respect to the training data set can be obtained. It is developed by Vladimir Vapnik and co-workers at AT&T Bell Laboratories in 1995.

The nonlinear SVMs are created by applying the kernel trick to maximummargin hyperplanes. The resulting algorithm is formally similar, except that every dot product is replaced by a nonlinear kernel function.

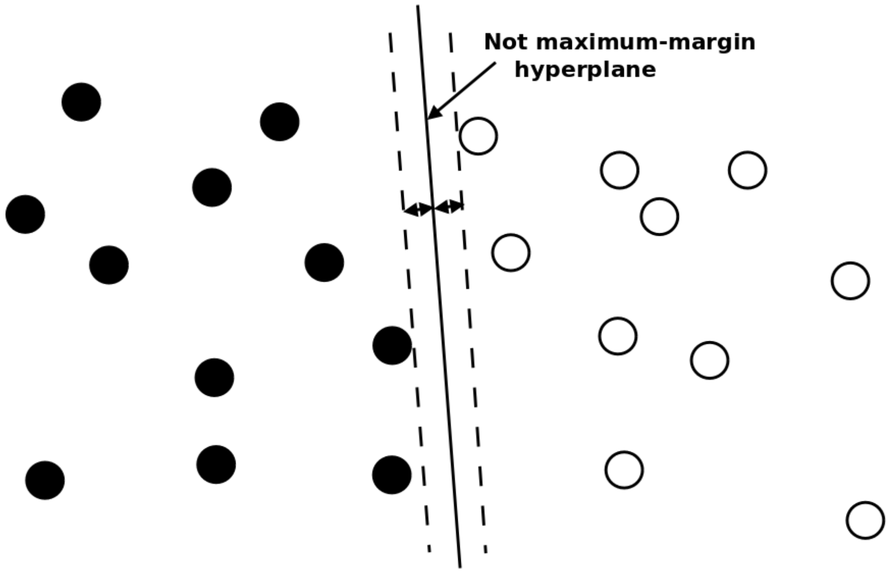


Figure 4.13: Misleading hyperplane

Kernel Function

The simplest way to divide two groups is with a straight line, flat plane or an N-dimensional hyper plane. But what if the points are separated by a non-linear region. In such case we would need a non-linear dividing line. Rather than fitting non-linear curves to the data, support vector machine handles this by using a kernel function to map the data into a different space where a hyperplane can be used to do the separation. [19] shows the use of second order polynomial kernel in the support vector machine. So, kernel function allows the algorithm to fit the maximummargin hyperplane in a transformed feature space. The trans-

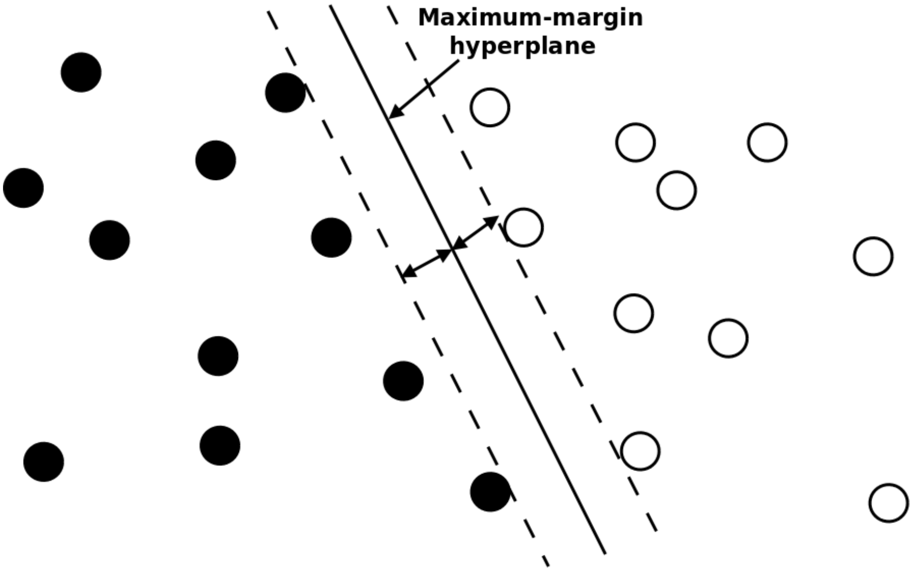


Figure 4.14: Support vector machine with maximum-margin hyperplane

formation may be nonlinear and the transformed space be high dimensional. For example, the feature space corresponding to Gaussian kernel is a Hilbert space of infinite dimension. Thus though the classifier is a hyperplane in the high dimensional feature space, it may be nonlinear in the original input space. Maximum margin classifiers are well regularized, so the infinite dimension does not spoil the result as the separation will be performed even with very complex boundaries.

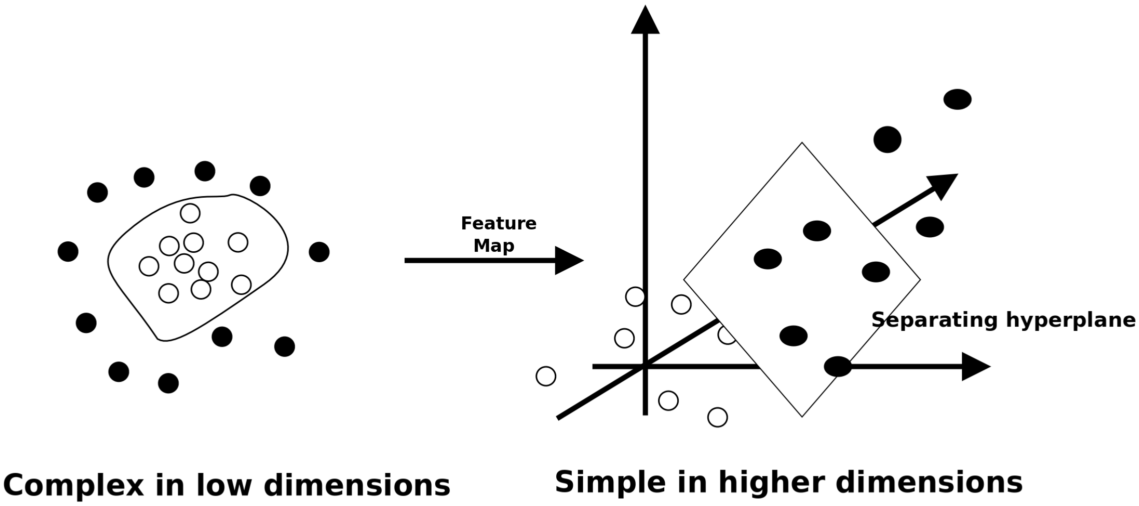


Figure 4.15: Support vector machine with kernel function

The effectiveness of SVM depends on the selection of kernel, the kernels parameters, and soft margin parameter c. Given a kernel, best combination of c and kernels parameters is often selected by a gridsearch with cross validation. The dominant approach for creating multiclass SVMs is to reduce multiclass problem into multiple binary classification problems. Common methods for such reduction is to build binary classifiers which distinguish between (i) one of the labels to the rest (oneversusall) or (ii) between every pair of classes (oneversusone). Classification of new instances for oneversusall case is done by a winnertakesall strategy, in which the classifier with the highest output function assigns the class. For the one versusone approach, classification is done by a maxwins voting strategy, in which every classifier assigns the instance to one of the two classes, then the vote for the assigned class is increased by one vote, and finally the class with most votes determines the instance classification. To tackle the same multiclass problem [25] has the utilization of DAG(Directed Acyclic Graph) SVMs in which a directed acyclic graph(DAG) of two-class SVMs is trained on each pair of classes in the data set.

5 SYSTEM DEVELOPMENT

5.1 Use Case Diagram

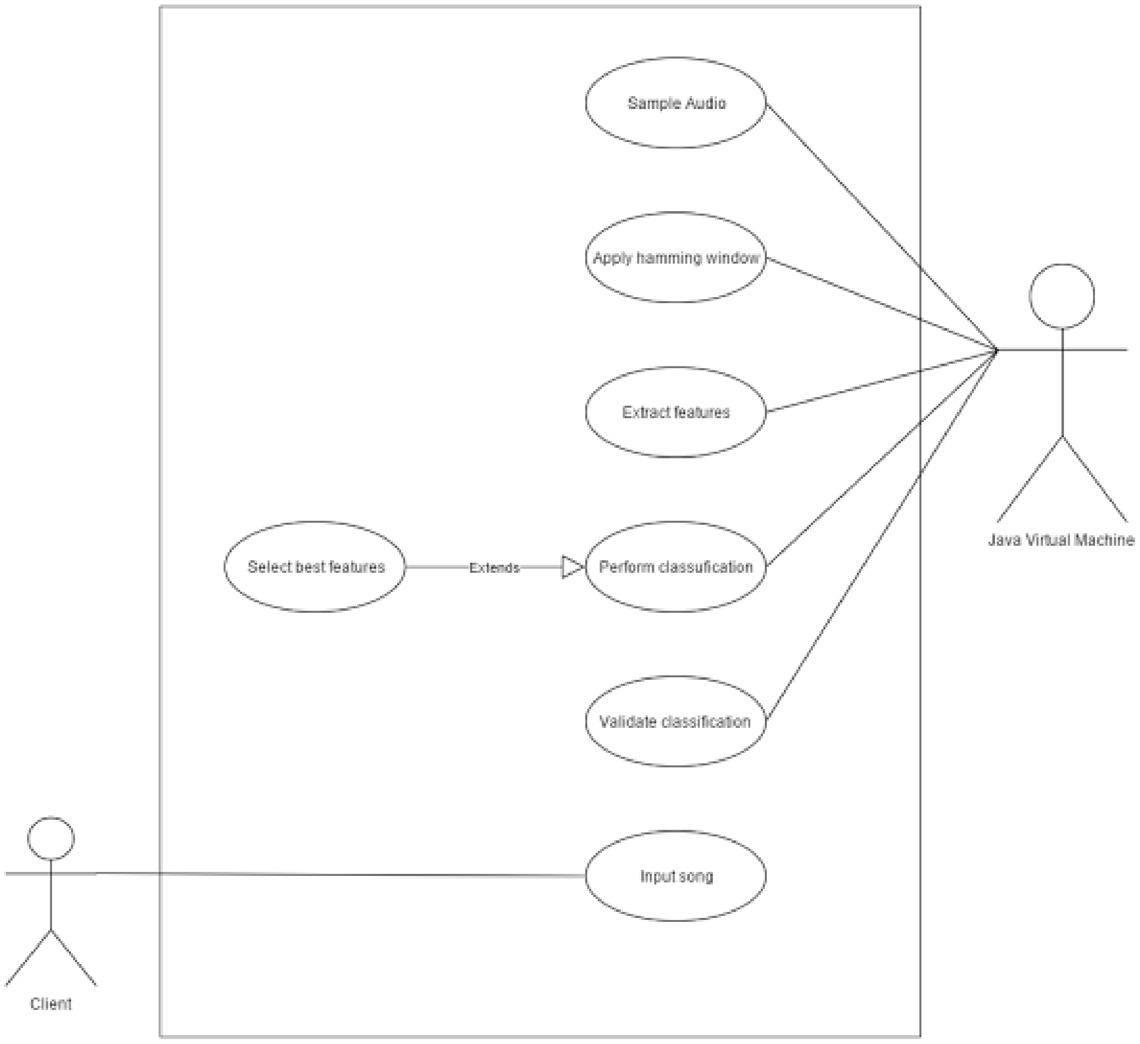


Figure 5.1: Use case diagram

5.2 Data Flow Diagram

5.2.1 Level Zero

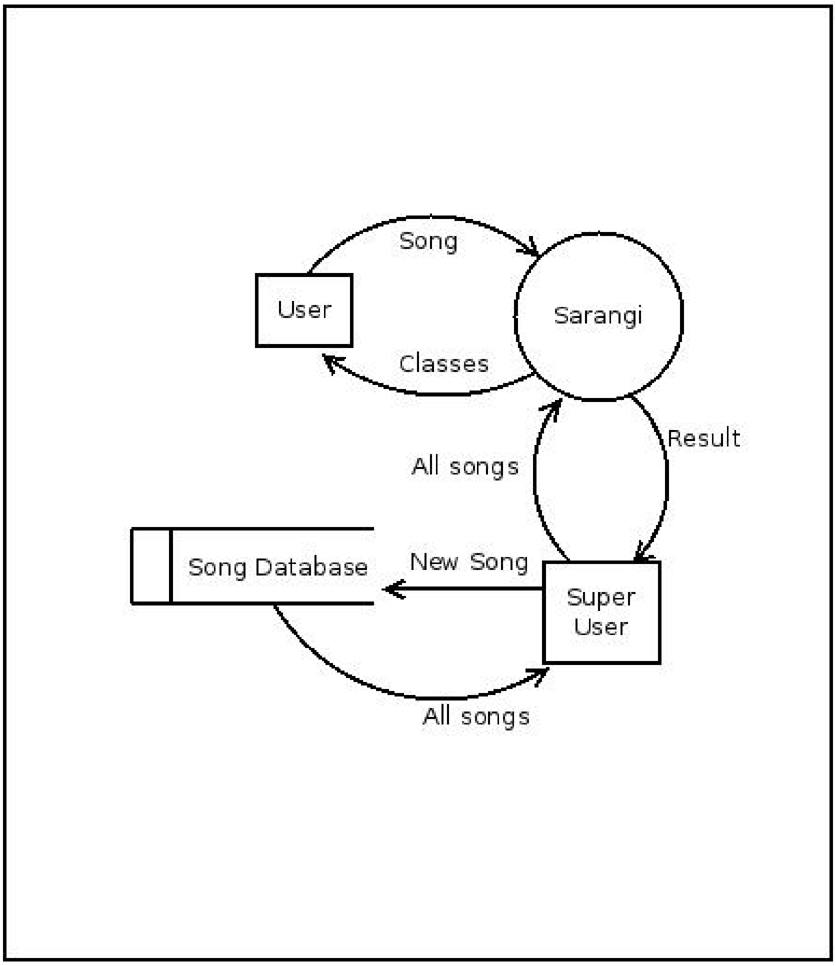


Figure 5.2: Context level zero

5.2.2 Level One

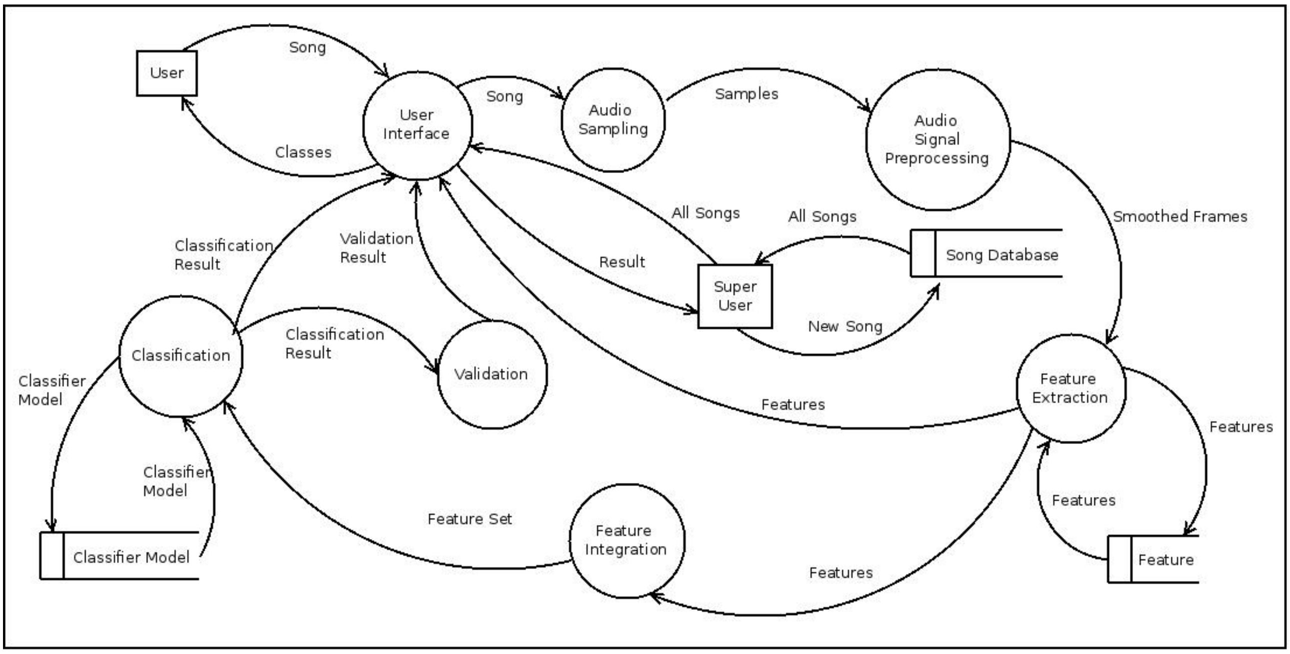


Figure 5.3: Context level one

5.2.3 Level Two

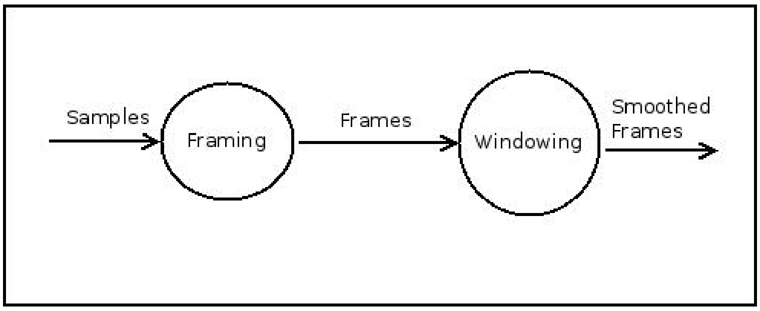


Figure 5.4: Audio signal pre-processing

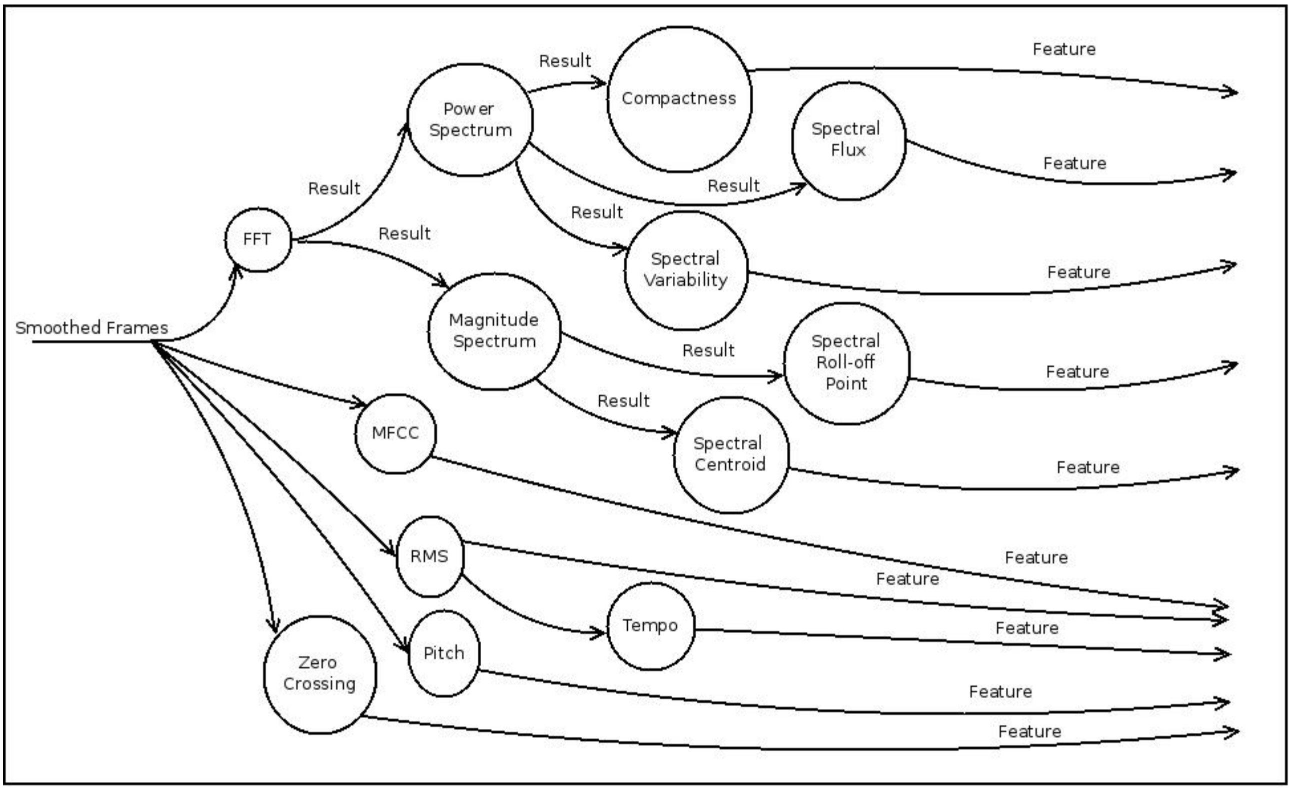


Figure 5.5: Feature extraction

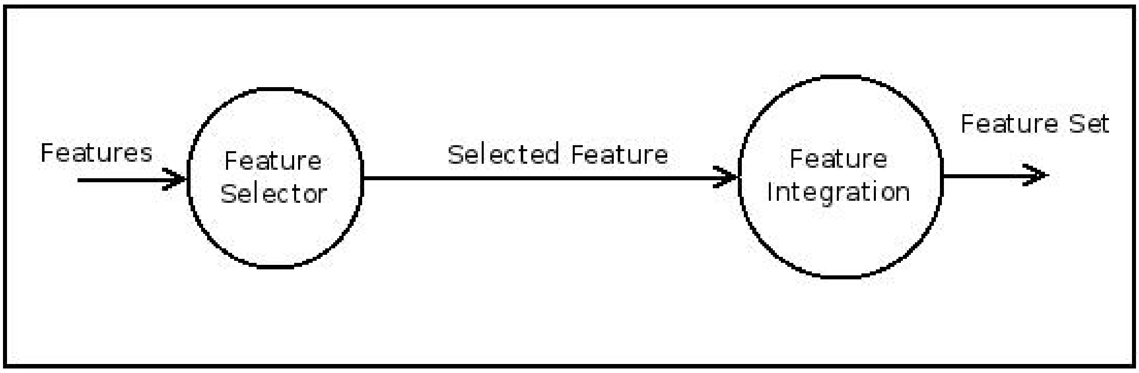


Figure 5.6: Feature integration

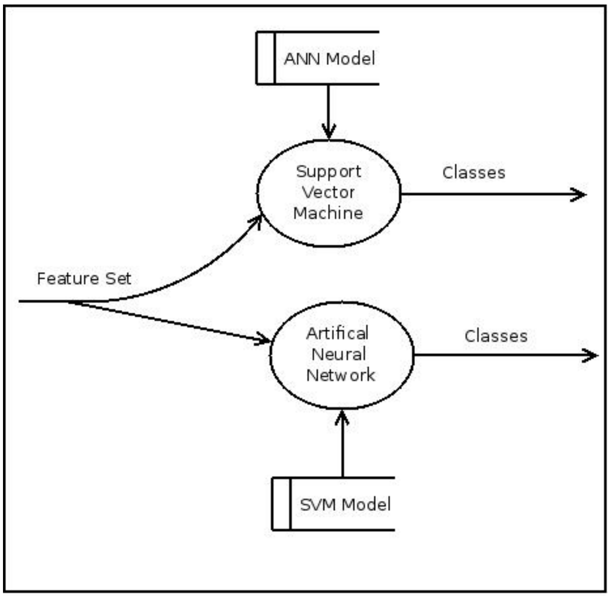


Figure 5.7: Classifier

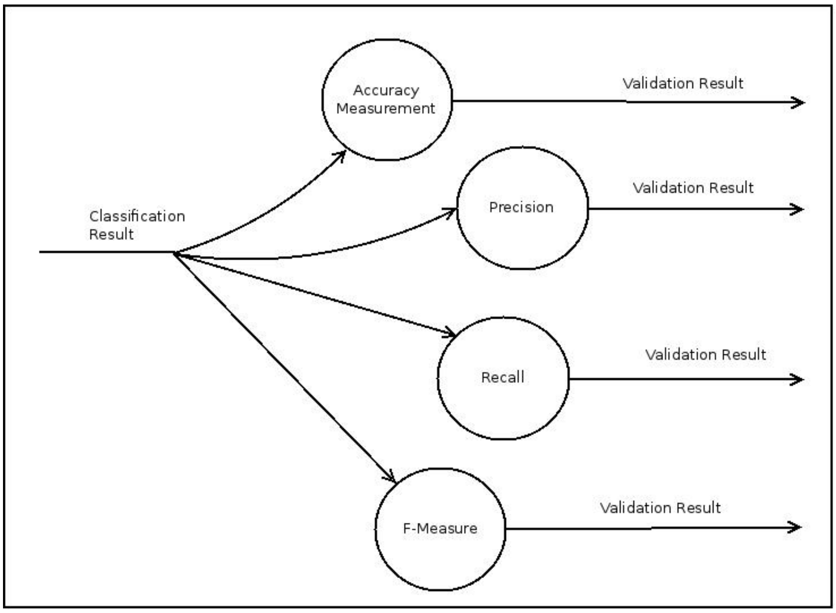


Figure 5.8: Validation

5.3 Sequence Diagram

5.3.1 Feature Extraction

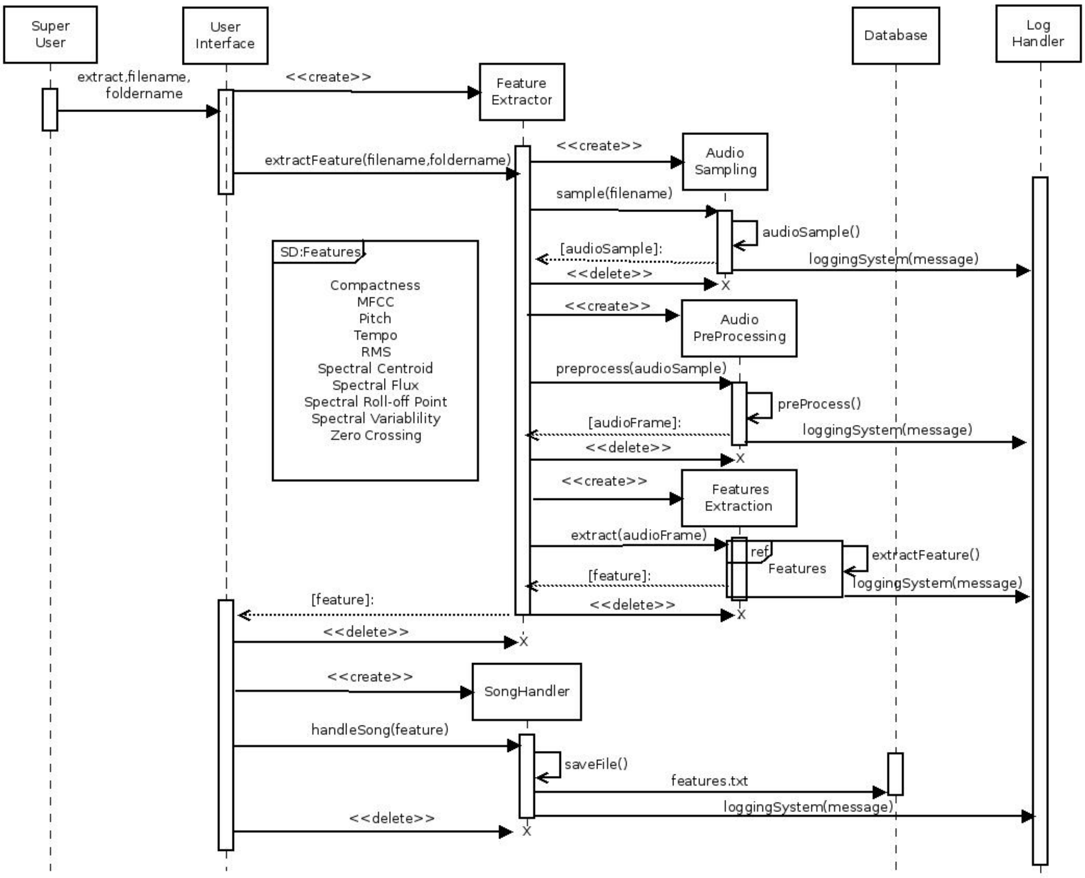


Figure 5.9: Sequence diagram of extracting feature

5.3.2 Classification

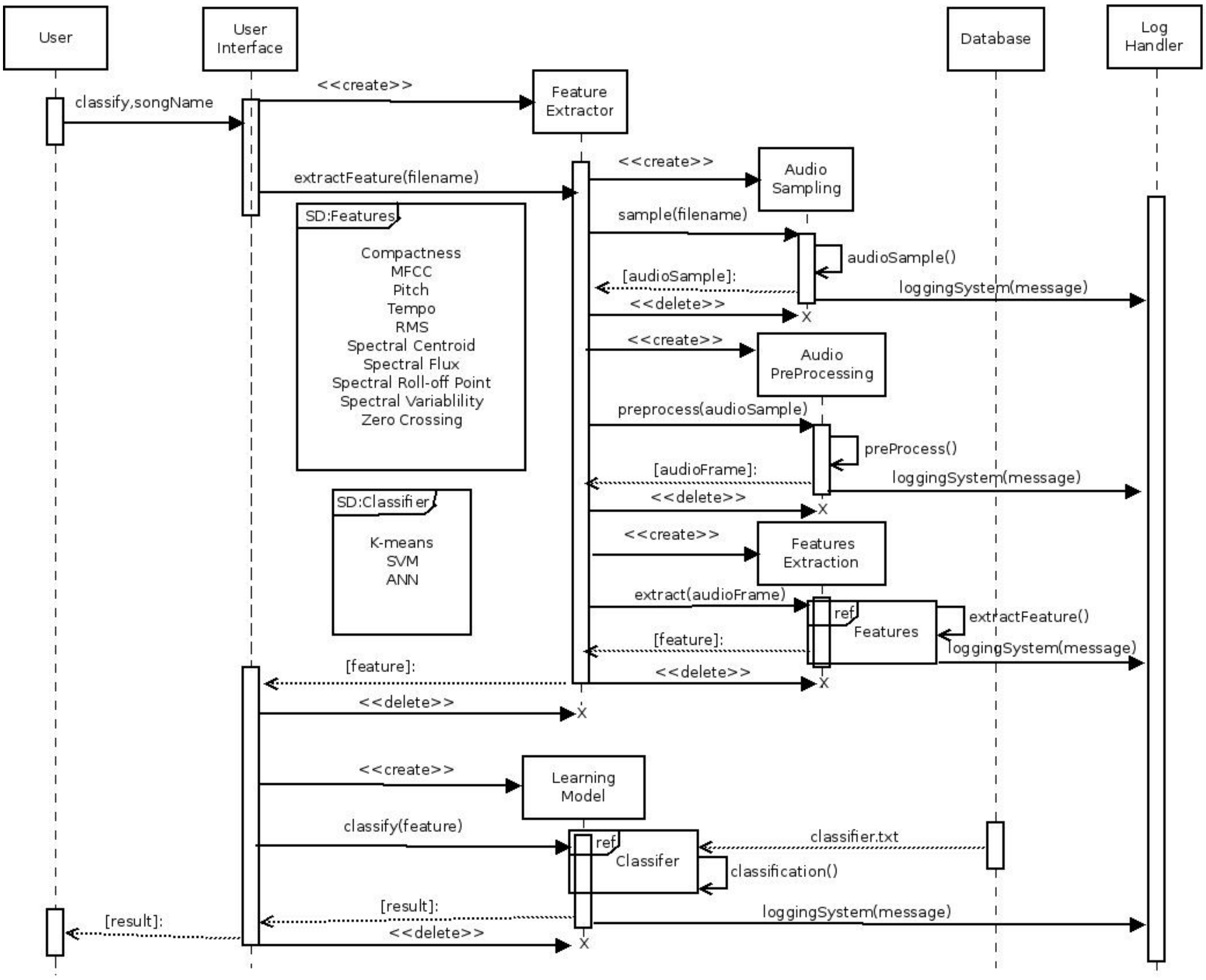


Figure 5.10: Sequence diagram of classification

5.3.3 Training

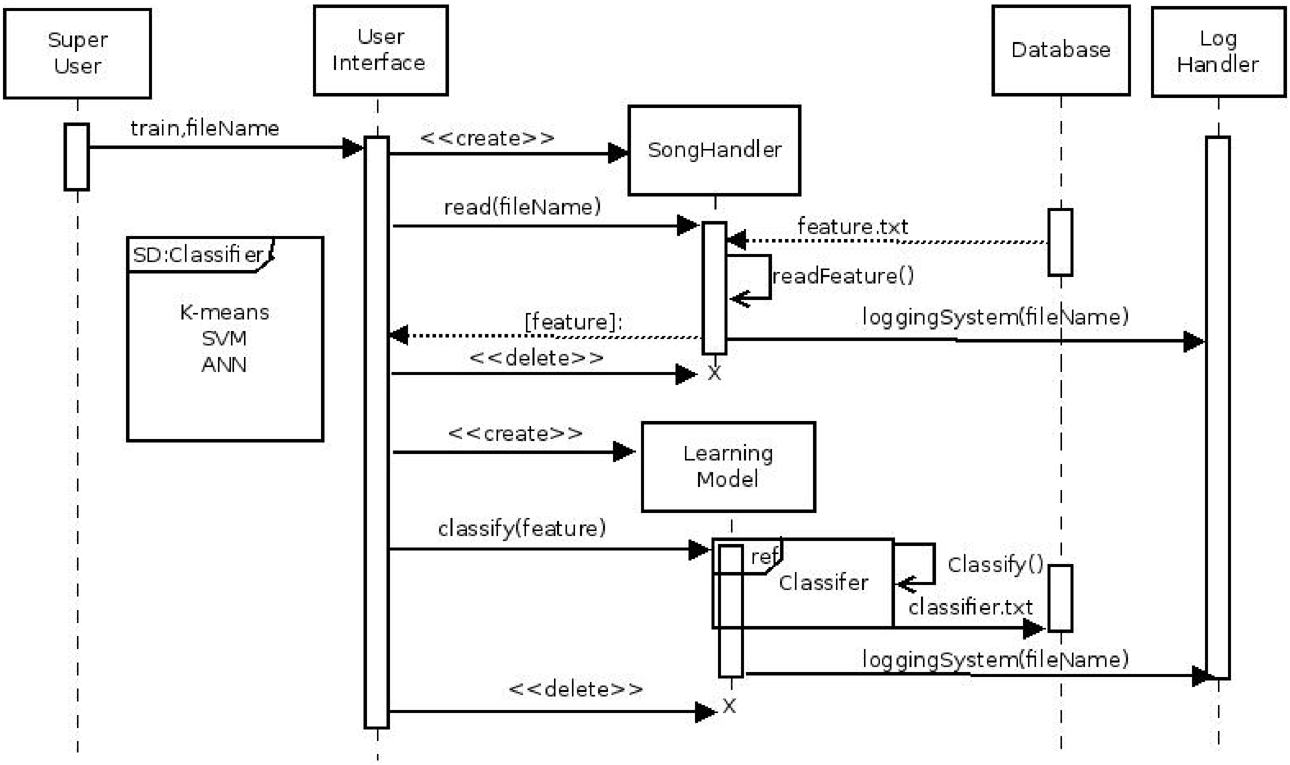


Figure 5.11: Sequence diagram of training

5.3.4 Testing

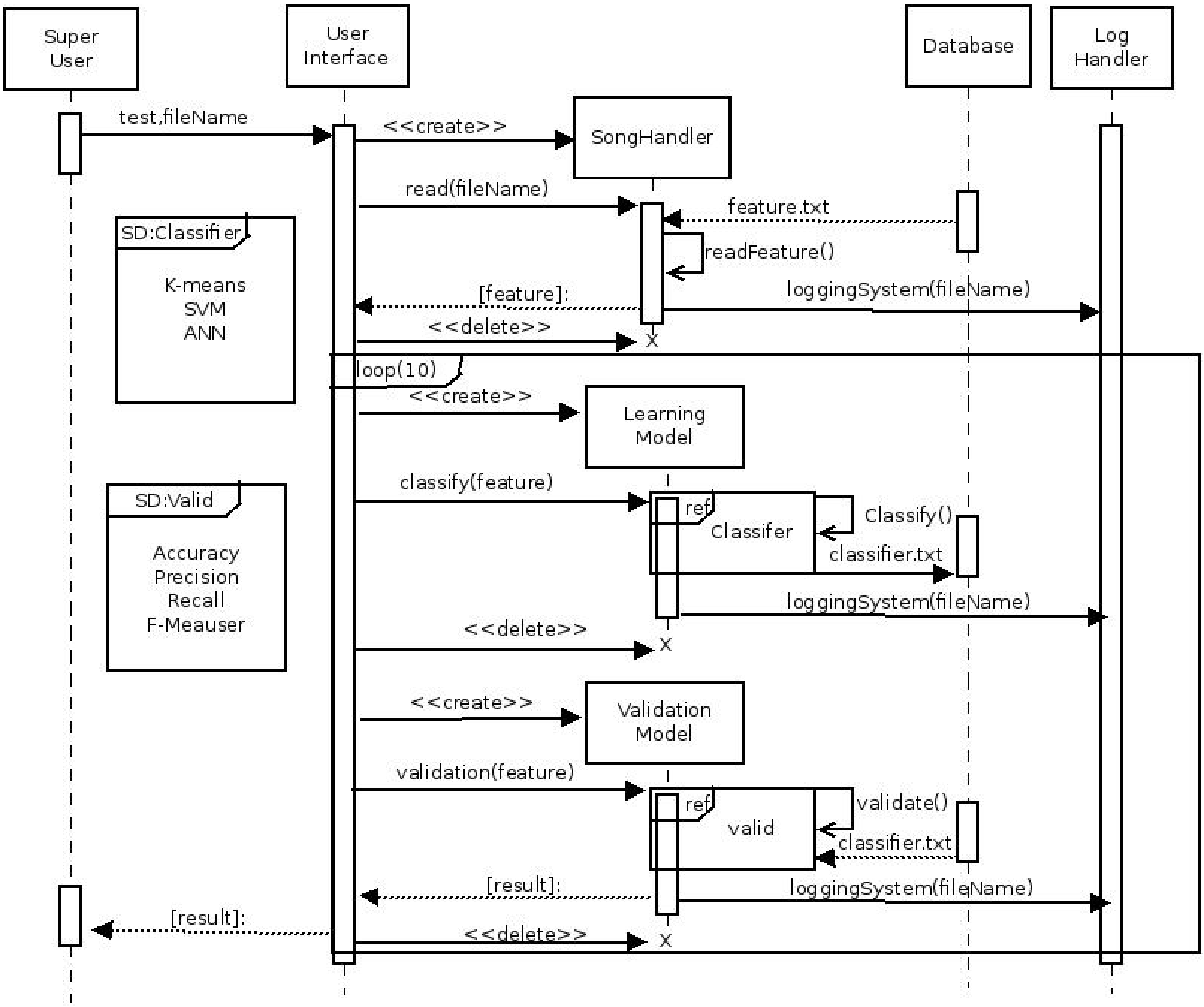


Figure 5.12: Sequence diagram of testing

5.4 Tools and Environment

5.4.1 Java Programming Language

Java is a general-purpose computer programming language that is concurrent class-based, object-oriented. Java language is used for overall development of the system. We chose java due to many reasons:

- We were familiar with Java having worked on pretty decent sized Java projects.

- Although Java may be too verbose for small applications, we believe it is
- Portability - Java runs well on all the popular platforms.
- Speed - The latest JIT compilers for Suns JVM approach the speed of C/C++ code, and in some memory allocation intensive circumstances, exceed it.
- Standard APIs that encompass lots of common funcionalities.
- GarbageCollection - the programmer doesn't have to worry about memory (most of the time).
- Availability of lots of Libraries, tools and frameworks in Java.

5.4.2 SMILE - Statistical Machine Intelligence and Learning Engine

SMILE (Statistical Machine Intelligence and Learning Engine) is a fast and comprehensive machine learning system. With advanced data structures and algorithms, Smile delivers state-of-art performance. A stable released version of SMILE 1.1.0 is used for implement of complex machine learning algorithm like SVM,etc. As mentioned in its documentation: "Smile covers every aspect of machine learning, including classification, regression, clustering, association rule mining, feature selection, manifold learning, multidimensional scaling, genetic algorithms, missing value imputation, efficient nearest neighbor search, etc."

5.4.3 TarsosDSP

TarsosDSP is a Java library used for audio processing. Its aim is to provide an easy-to-use interface to practical music processing algorithms implemented, as simply as possible, in pure Java and without any other external dependencies. The library tries to hit the sweet spot between being capable enough to get real tasks done but compact and simple enough to serve as a demonstration on how DSP algorithms works. We used TarsosDSP to process the low level song features.

5.4.4 Google Gson

Gson is a Java library that can be used to convert Java Objects into their JSON representation. It can also be used to convert a JSON string to an equivalent Java object. Gson can work with arbitrary Java objects including pre-existing objects that you do not have source-code of. We used Gson to store the features as JSON and read them back. We chose JSON because it is easily readable through a lot of methods, libraries and languages.

5.4.5 Xstream

XStream is a simple library used to serialize objects to XML and back again. We used XStream to store the model as a XML file so that it can be read back.

5.4.6 MP3SPI

MP3SPI is a Java Service Provider Interface that adds MP3 (MPEG 1/2/2.5 Layer 1/2/3) audio format support for Java Platform. We used it to add a more reliable support for mp3 files.

5.4.7 Python Programming Language

Python is a high-level, general-purpose, interpreted, dynamic programming language. We used Python scripts to handle the audio files and modify them to create the mood based dataset.

6 RESULTS AND ANALYSIS

6.1 Dataset pre-processing

6.1.1 Genre Dataset

In [28], Downie presents the problem of a lack of a proper baseline dataset for the AMGC problem, which meant research teams couldnt scientifically compare and contrast their different approaches.

However, the publicly available GTZAN dataset introduced in [12], solved most parts of the issue by soon becoming the standard dataset used by researchers across the world. We too used this dataset for this project. The dataset contains 100 representative excerpts from ten different genre. They were taken from radio, compact disk, and MP3 compressed audio files. All the files are stored as 22 050 Hz, 16-bit, mono audio files. The Genres dataset has the following classes: classical, country, disco, hiphop, jazz, rock, blues, reggae, pop, metal. Among them we were only concerned with five genre as mentioned in Chapter 1: classical, hiphop, jazz, rock, and pop, So, we had a total of 500 songs in our dataset.

6.1.2 Mood Dataset

Due to the high variability in the emotional models adopted (whether categorical or dimensional), there have been no popular baseline dataset for MER task.

So, to tackle the issue, in 2013, Soleymani et al. [36] created a 1000 songs dataset for emotional analysis of music which uses the Valence-Arousal axes for representing emotional values for songs.

The songs, in the dataset, each 45 seconds long, were collected from FMA. They used Amazon Mechanical Turk as a crowdsourcing platform for collecting more than 20,000 annotations on the 1,000 songs.

Furthermore, their analysis on the annotations revealed a higher agreement in arousal ratings compared to the valence ratings.

We have used a filtered version (with some redundancies removed) of that dataset resulting in a final set of 744 songs. We further labeled them as high/low arousal and high/low valence songs based on the numerical values in the dataset. To achieve equal number of songs in each class, we finally used 600 songs of those 744 songs.

6.2 Testing and Validation

While building a software product, the concept of testing and validation plays an important role throughout its development. Along with the development of a software product, numerous testing and validation are need to be performed regulary. In fact, most of the development period is given to testing and validation. It's important as it conforms to whether we are building product right and whether we are building the right product. After series of testing and validation only we can point that the system conforms to the desired specifications and need.

Since our software project is also research oriented, so we need series of rigorous testing. Not only this, being a predictive system/model a type of validation and measure for performance is a must, so that it can analyze the result and decide how to progress further accounting the fact. For our system, we chose cross-validation after we confronted it's popularity for a predictive model and ease of use. Similarly for measure of performance we chose recall, presicion and F-measure.

6.2.1 Cross-validation

Cross-validation is a technique for estimating the performance of a predictive model. Cross-validation, sometimes called rotation estimation is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set. In a predictive problem like our project, a model is usually given a dataset of known data(training dataset) on which the model is trained, and a dataset of unknown data (or first seen data) against which the model is tested (testing dataset). The goal of cross validation is to define a dataset to ”test” the model in the training phase(validation dataset), in order to limit problems like overfitting, give an insight on how the model will generalize to an independent dataset(and unknown dataset, for instance from a real problem),etc.

One of the main reasons for using cross-validation instead of using the conventional validation is that there is not enough data available to partition it into separate training and test sets without losing significant modelling or testing capability. There are several cross-validation method out there like leave-p-out cross-validation, leave-one-out cross validation,2-fold cross validaiton, repeated random sub-sampling validation, etc. But we decide to got with k-fold cross validation. The main reason for this selection is that k-fold cross-validation estimator can prove to have a lower variance than a single hold-out set estimator if the amount of data available is limited. Our main goal with k-fold cross validation is to estimate the expected level of fit of a model to a dataset that is independent of the data that were used to train the model.

In k-fold cross-validation, the original sample is randomly partitioned into k equal sized subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k 1 subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged to produce a single estimation. The advantage of this method over repeated random sub-sampling (see below) is that all observations are used for both training and validation, and each observation is used for validation exactly once. ten-fold cross-validation is currently applied in our system. Our music data set is first divided into ten equal samples of songs, and on every run for ten times, a different sample of songs are used for testing and the rest for training.

This k-fold validation validation process is applied with each performance metrics and the appropriate features are taken to classify in our system.

6.2.2 Measure of Performance

In a predictive problem, we always need a measure of performance for that predictive model along with cross validation. Based on the value of different attributes falling under the measure of performance, we can then have the knowledge of how well the system is performing. It is important because as a matter of fact it provides much deeper insight of the performance of the predictive model, not only accuracy.

Our measure of performance include precision, recall and F-measure. We can say our measure of performance are based on the relevant and non-relevent documents. In this case, relevant documents are those one which simply belong to the relevant category. Relevant category is also called the true positives. True positive is the number of items correctly labeled as belong to the positive class or say which are predicted correctly. The not relevant documents are those one which simply are not relevant to the category or say they are false negative. False negative is the number of negative items which are predicted to fall under the positive class. False Positive is the number of negative items which are predicted to be in true class. Similarly, true negative is the number of negative items which are correctly predicted to fall under negative class.

- **Precision**

  The precision of a measurement system, related to reproducibility and repeatability, is the degree to which repeated measurements under unchanged conditions show the same results. In a classification task, the precision for a class is the number of true positives divided by the total number of elements labeled as belonging to the positive class. A perfect precision score of 1.0 means that every result retrieved by a search was relevant which means falling to the specified class. But precision does not say whether all items of that class were correctly predicted.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{26}$$

- **Recall**

  Recall literally is how many of the true positives were recalled (found), i.e. how many of the correct hits were also found. Recall is also sometimes called sensitivity or probability of detection. It is the true positive rate as it measure the proportion of positives that are correctly identified as such. Recall is the number of true positives divided by the total number of elements that actually belong to the positive class. A perfect recall score of 1.0 means that all relevant documents were retrieved by search. We can also say that perfect recall shows that all items of a class were labeled fall under same class. But recall does not say that how many other items were misclassified to fall under same class.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{27}$$

- **F measure** A measure that combines precision and recall is the harmonic mean of precision and recall. F measure is the weighted harmonic mean of its the precision and recall of a system. It is sometimes called as balanced F-score. This measure is approximately the average of the two when they are close, and is more generally the harmonic mean which for the case of two numbers coincides with the square of the geometric mean divided by the arithmetic mean. There are several reasons that the F-score can be criticized in particular circumstances due to its bias as an evaluation metric. This is also known as the F_1 measure, because recall and precision are evenly weighted.

$$F = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{28}$$

All three performance metrics have been used to determine the best features to be used for classification in cases of genre, arousal and valence.

6.3 Audio Pre-processing

6.3.1 Effect of Window Functions

We tried out five window functions to smoothen the data frames obtained from the samples.

As seen from the figure, Hanning window gave the best performance.

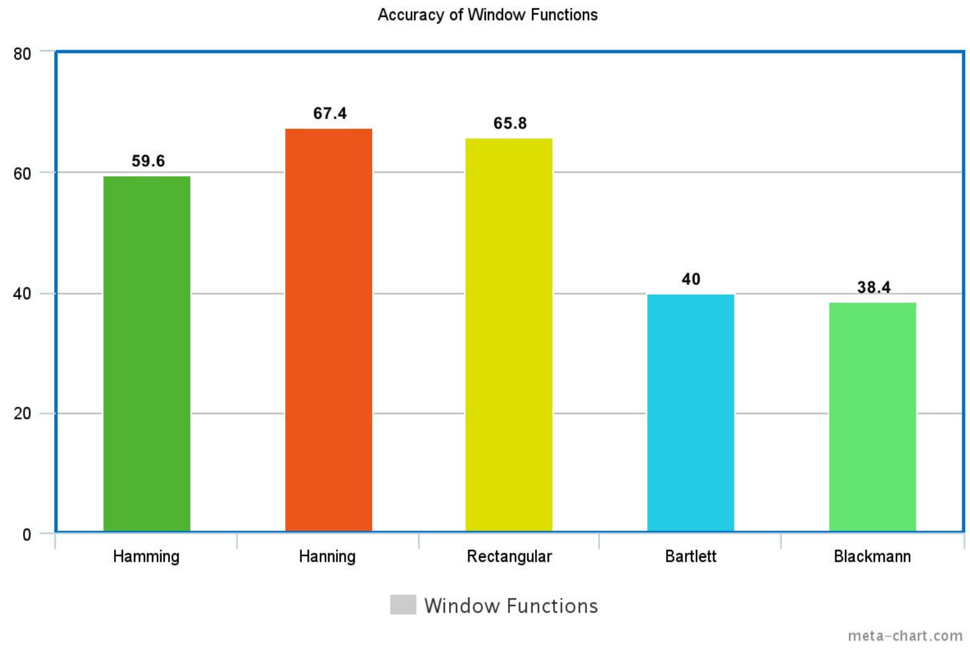


Figure 6.1: Effect of window functions

6.3.2 Effect of Frame Size

The Frame-size needs to be small enough to be statistically invariant but big enough to capture sufficient information. Generally, 20-40ms is taken as the frame size. We tested the frame size according to the number of samples per frame: 512, 1024, 1536 and 2048 samples which correspond to roughly 11.5ms, 23ms, 34.5ms and 46 ms for our songs which have a sampling rate of 44.1 KHz.

As seen in the figure, frame size of 11.5ms and 23ms performed considerably better than the bigger frame sizes. We chose the 23ms (1024 samples) frame size because the smaller 512 sample frame size would lead to higher number of frames and hence necessitate more computation.

6.3.3 Effect of Frame Overlap

We explored four different overlapping schemes: zero per cent, 25 per cent, 50 per cent and 75 per cent overlap.

In each of the cases, we received almost the same accuracy (75.4 per cent on No-overlap, 75.8

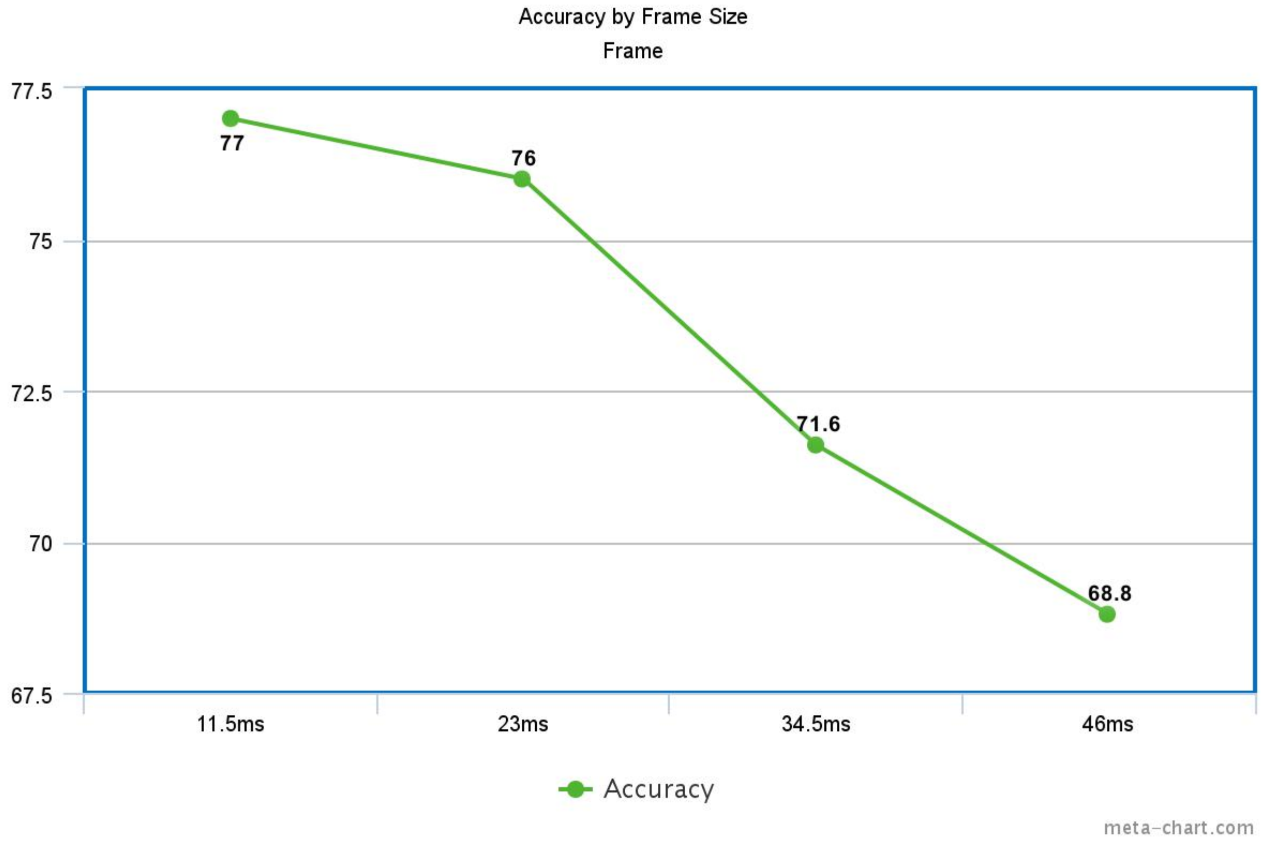


Figure 6.2: Effect of frame size

per cent on Quarter overlap, 76 per cent on half-overlap, and 75.2 per cent on three-quarters overlap). And so, as it seemed to indicate that the overlapping didnt have any bearing on our results, we chose the less computationally intensive option of using no overlap at all.

6.3.4 Effect of Pre-Emphasis

Pre-emphasis decreased the accuracy from 75.4 per cent to 62.6 per cent.

6.4 Analysis of Features

6.4.1 Genre Based Classification

Table 6.1: Genre classification using ANN

Feature	Classical	Hiphop	Jazz	Pop	Rock	Overall
Spectral Centroid	47.54	11.92	11.61	51.25	18.89	28.40
MFCC	92.45	83.42	91.57	82.98	74.44	84.20
Zero Crossing	63.29	48.31	39.83	52.65	52.00	51.20
Pitch	37.83	15.00	34.80	61.73	1.67	28.00
Compactness	81.53	81.75	57.67	28.39	45.55	58.60
Timbre	6.25	20.00	30.00	20.00	28.46	20.60
RMS	85.52	34.99	19.85	47.89	70.46	51.20
Spectral Flux	87.94	26.71	19.77	43.63	57.45	46.40
Spectral Roll off point	84.10	54.11	22.14	43.74	13.18	41.60
Spectral Variability	83.10	32.98	25.98	51.24	71.76	52.40

Table 6.2: Genre classification using SVM

Feature	Classical	Hiphop	Jazz	Pop	Rock	Overall
Spectral Centroid	58.95	1.11	69.09	7.50	0.00	26.40
MFCC	91.79	85.25	87.98	85.62	77.61	85.80
Zero Crossing	63.20	48.86	41.47	58.51	44.31	50.00
Pitch	59.95	38.37	36.04	35.77	13.64	36.20
Compactness	67.18	66.13	42.30	47.93	53.60	55.60
Timbre	1.67	37.83	34.80	61.73	15.00	28.00
RMS	20.00	40.00	30.00	20.00	0.00	21.60
Spectral Flux	59.17	10	20.91	33.06	0.00	27.20
Spectral Roll off point	34.34	52.60	29.07	14.29	0.00	24.40
Spectral Variability	28.46	20.00	30.00	20.00	6.25	20.60

6.4.2 Mood Based Classification

Table 6.3: Mood classification(Arousal) using ANN

Feature	Low Arousal	High Arousal	Overall
Spectral Centroid	70.07	26.20	50.34
MFCC	69.32	75.09	71.38
Zero Crossing	70.70	64.03	67.24
Pitch	44.27	64.62	54.83
Compactness	59.73	51.71	57.24
Timbre	58.96	61.10	58.28
RMS	70.76	67.28	68.97
Spectral Flux	76.58	58.87	67.93
Spectral Roll off point	50.79	47.67	50.34
Spectral Variability	73.28	62.51	67.24

Table 6.4: Mood classification(Arousal) using SVM

Feature	Low Arousal	High Arousal	Overall
Spectral Centroid	50.00	50.00	44.14
MFCC	73.22	71.77	72.41
Zero Crossing	74.00	67.07	70.69
Pitch	59.00	55.01	56.55
Compactness	47.22	78.45	62.76
Timbre	62.58	53.87	56.55
RMS	50.00	50.00	42.76
Spectral Flux	40.00	60.00	45.86
Spectral Roll off point	70.00	30.00	41.38
Spectral Variability	50.00	50.00	40.00

Table 6.5: Mood classification(Valence) using ANN

Feature	Low Valence	High Valence	Overall
Spectral Centroid	37.69	60.63	51.38
MFCC	60.04	65.45	63.79
Zero Crossing	62.74	57.80	59.66
Pitch	66.29	35.46	50.69
Compactness	50.67	58.79	52.76
Timbre	61.75	57.69	56.90
RMS	63.85	44.87	53.79
Spectral Flux	70.19	47.50	59.31
Spectral Roll off point	57.56	44.94	48.28
Spectral Variability	63.99	45.13	51.03

Table 6.6: Mood classification(Valence) using SVM

Feature	Low Valence	High Valence	Overall
Spectral Centroid	40.00	60.00	44.83
MFCC	45.37	72.06	58.62
Zero Crossing	70.34	52.46	60.34
Pitch	53.56	52.53	49.66
Compactness	57.25	62.40	58.97
Timbre	60.50	61.19	60.34
RMS	50.00	50.00	44.83
Spectral Flux	60.00	40.00	40.34
Spectral Roll off point	60.00	40.00	41.72
Spectral Variability	50.00	50.00	41.83

6.4.3 Effect of Number of MFCCs on Result

The results indicate that once we use more than 10 MFCC Coefficients, the accuracy plateaus and doesnt increase at all. So, using around 15 coefficients is found to be good enough for the problem domain.

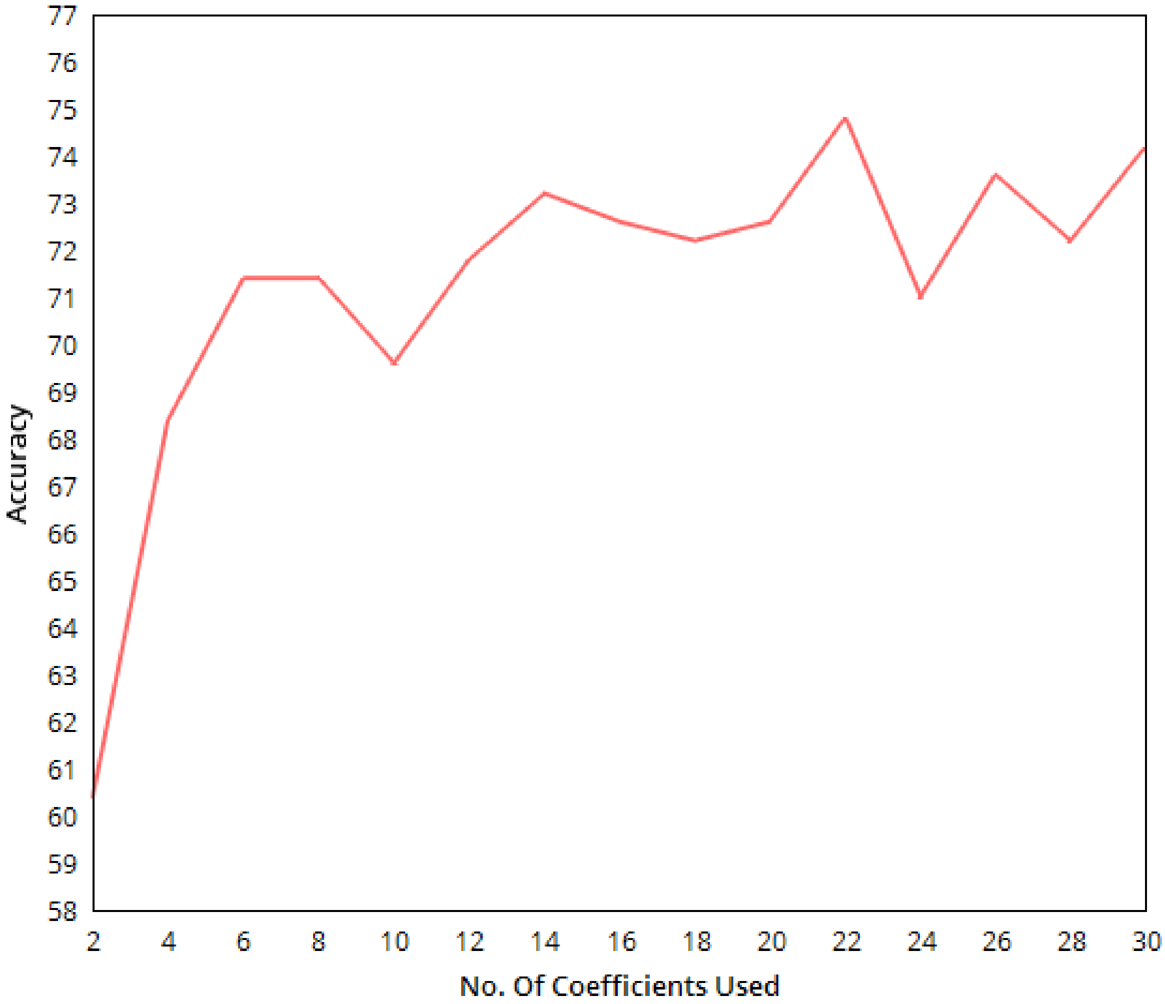


Figure 6.3: Effect of number of MFCCs on result

6.5 Analysis of Classifiers

6.5.1 SVM

6.5.1.1 Selection of Kernels. Linear Kernel performed best for the Genre Classification Problem and so we selected it for Genre Classification.

As for the Mood Classification problem, both Gaussian and Laplacian out-performed the Linear Kernel in both arousal and valence classification. So, we selected Laplacian Kernel (based on its slightly superior performance) as our Kernel for Mood Classification.

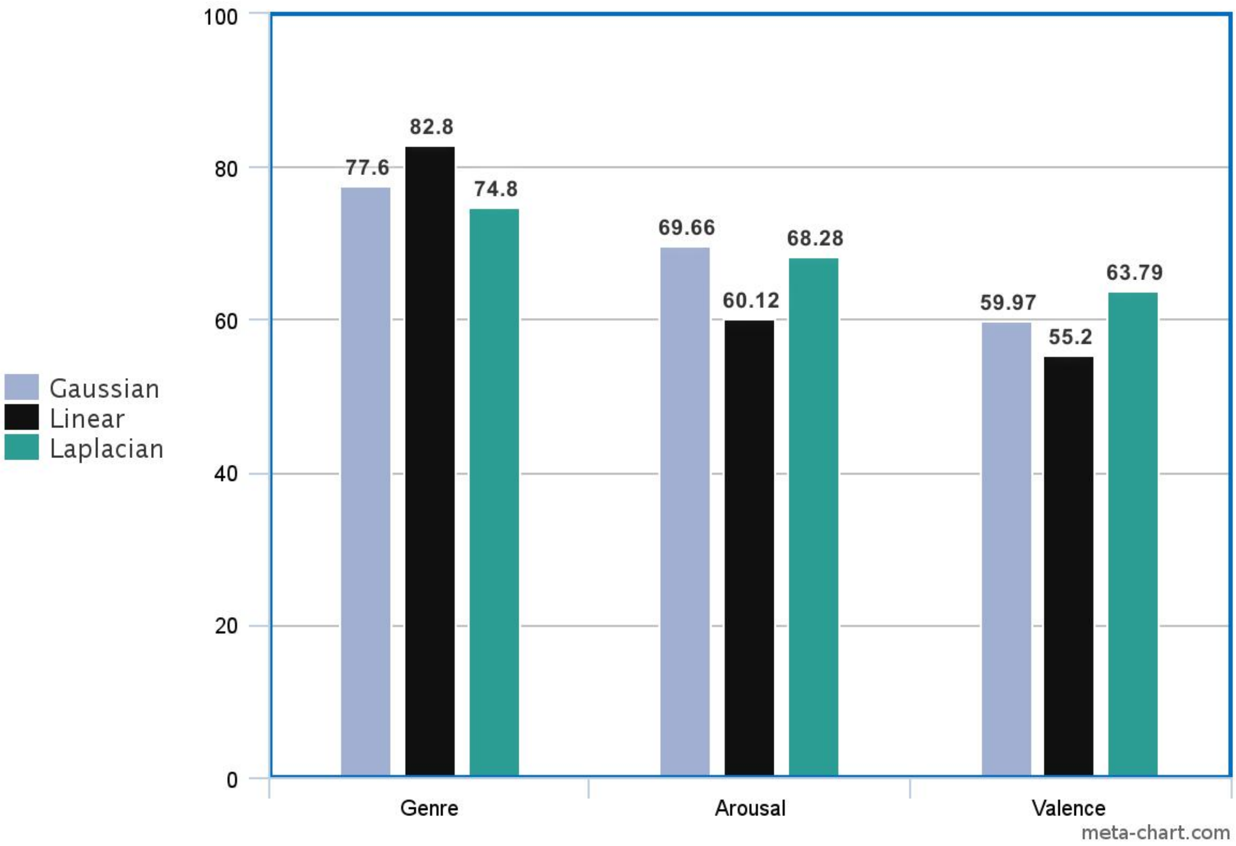


Figure 6.4: SVM Kernel

6.5.1.2 Effect of Learning Iterations. We tested three iteration values to train SVM: one, 10 and 100.

For Genre Classification, all three of them resulted in a similar performance of around 77 per cent accuracy.

For Arousal, the performance increased when going from one (61 per cent) to 10 (69 per cent) iterations but increasing it further to 100 iterations didnt increase the performance. We still received an accuracy of around 69 per cent.

6.5.2 ANN

6.5.2.1 Activation and Error Functions. We tried out two sets of Activation and Error Functions for the Neural Network with a 1000 iteration training.

The results are given below:

Table 6.7: Genre

Functions	Accuracy(percent)	Precision	Recall
ross Entropy Error with Softmax Activation	85.80	0.92	0.92
Least Mean Squares Error with Logistic Sigmoid Activation	83.60	0.87	0.89

So, the first combination slightly outperforms the second in all three metrics.

Table 6.8: Arousal

Functions	Accuracy(per cent)	Precision	Recall
Cross Entropy Error with Softmax Activation	69.31	0.72	0.64
Least Mean Squares Error with Logistic Sigmoid Activation	70.34	0.72	0.65

Table 6.9: Valence

Functions	Accuracy(per cent)	Precision	Recall
Cross Entropy Error with Softmax Activation	63.27	0.67	0.60
Least Mean Squares Error with Logistic Sigmoid Activation	64.31	0.68	0.60

Here, the two combinations are almost inseparable in terms of performance.

6.5.2.2 Number of Nodes in Hidden Layer.

We used only one hidden layer as it should be enough for our problem domain. As seen in the figure, for any number of hidden numbers after six or so, we get almost the same accuracy. As a rule of thumb, it is usually

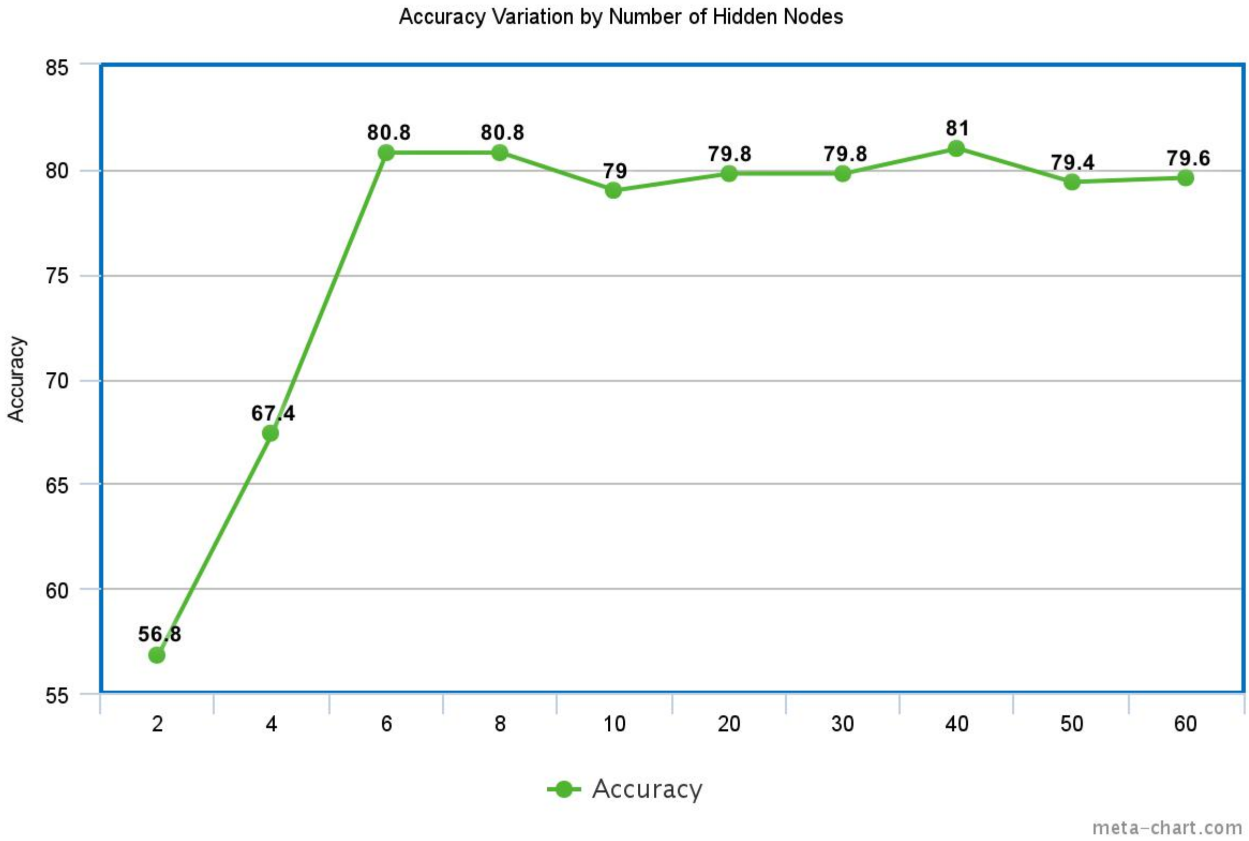


Figure 6.5: Effect of number of hidden nodes

recommended that the number of nodes be around the mean of the number of inputs and outputs, so we chose 30 as our final number of hidden nodes.

The number of nodes had minimal effect in regard to mood classification.

6.5.2.3 Number of Iterations.

As seen in the figure, for genre classification, the number of iterations has an effect on the accuracy up to a certain point (around 20 iterations).

As for Arousal, the increase in iterations had no effect on the accuracy.

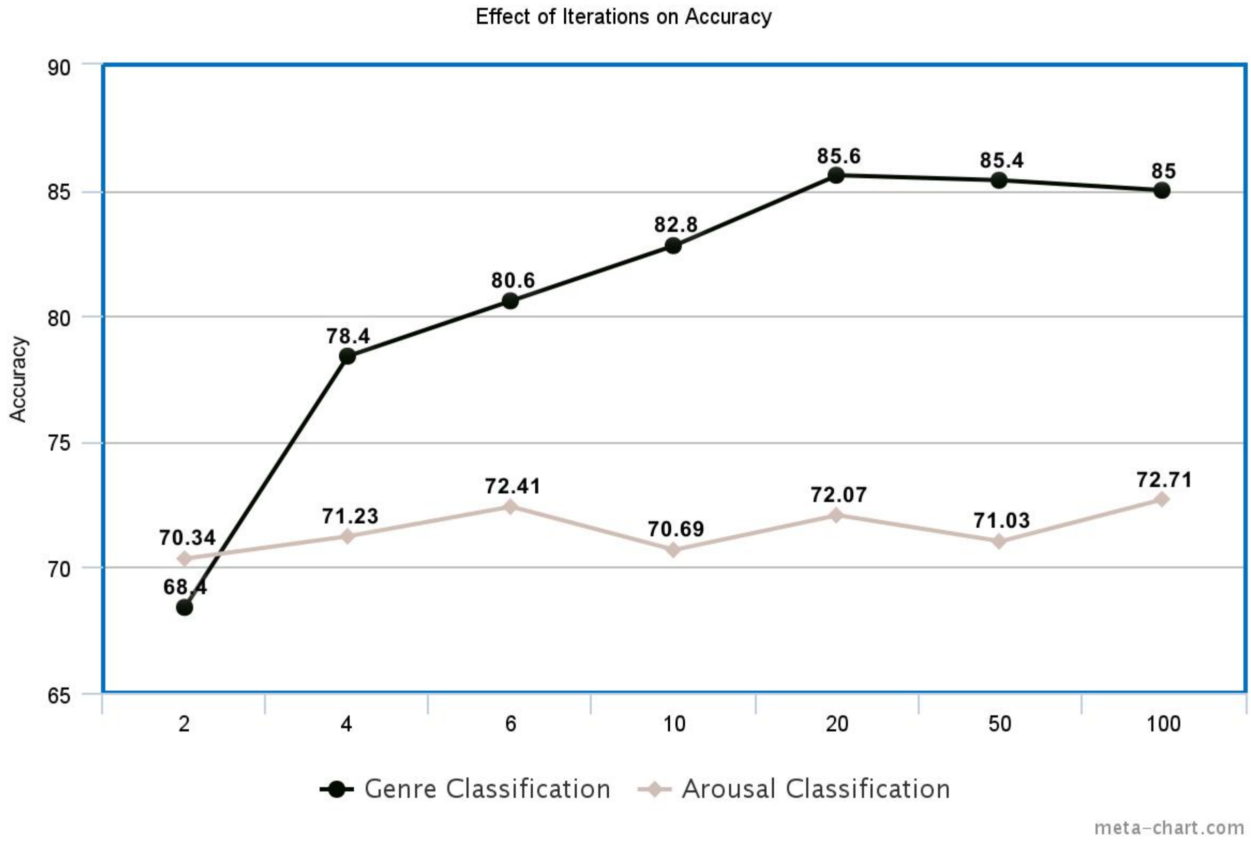


Figure 6.6: Effect of number of iterations

6.5.2.4 Learning Rate.

A learning rate of 0.1 was enough to get a decent amount of accuracy. A lower rate than that affected the accuracy while a higher rate didnt have any influence at all.

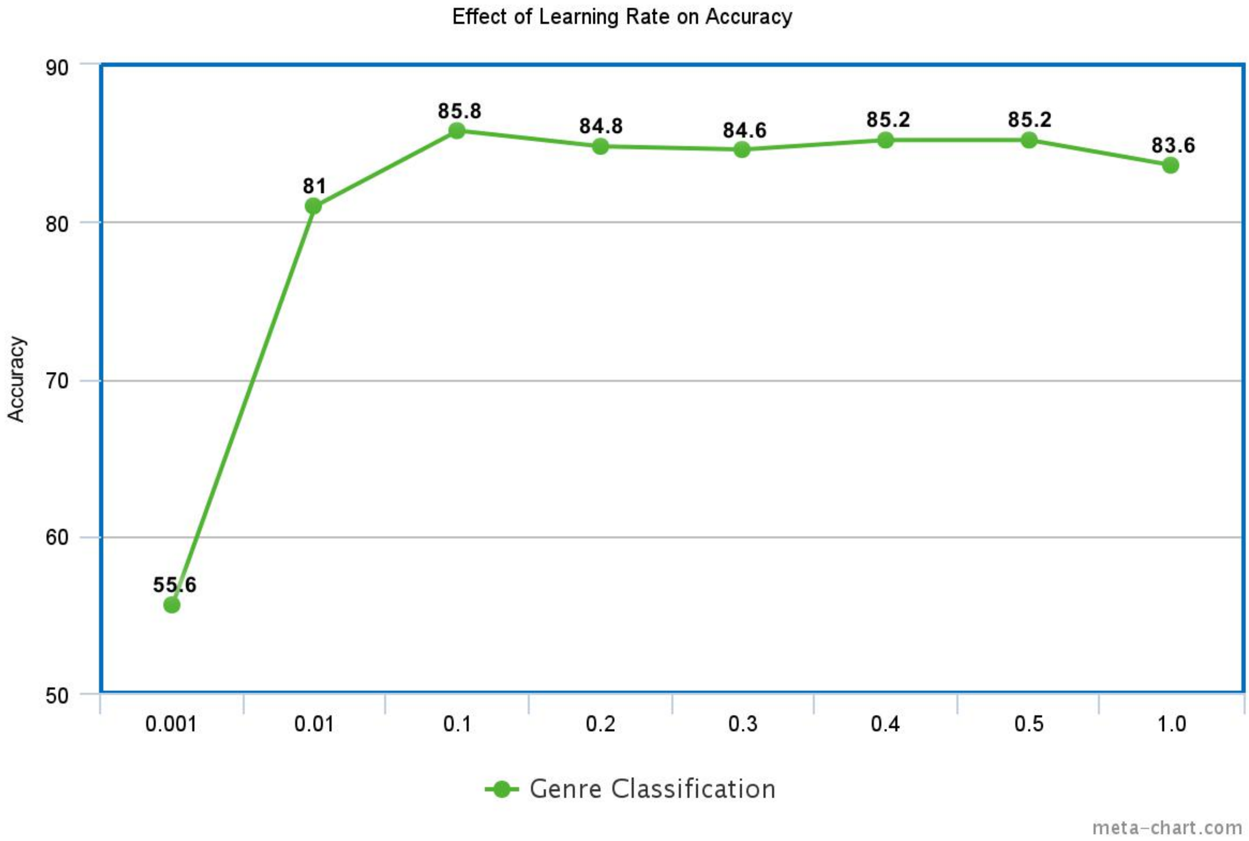


Figure 6.7: Effect of learning rate

6.6 Final Results

6.6.1 Genre Classification

Table 6.10: Genre classification performance measure

Classifier	Precision	Recall	F-Measure	Accuracy
SVM	0.87	0.94	0.89	83.00
ANN	0.94	0.92	0.92	88.80

For our final model we used ANN with these feature: MFCC, Spectral Centroid, Zero Crossing, Compactness and RMS.

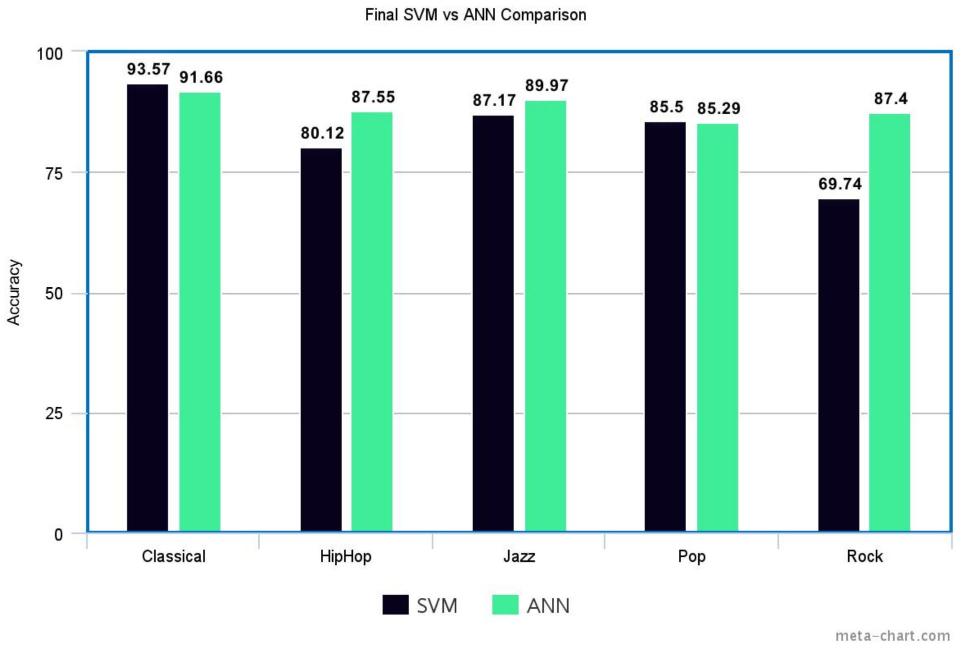


Figure 6.8: Final SVM and ANN comparision based on genre

Table 6.11: Genre classification confusion matrix

	Classical	Hiphop	Jazz	Pop	Rock
Classical	93	1	4	0	2
Hiphop	0	88	0	3	9
Jazz	5	0	90	1	4
Pop	2	7	0	86	5
Rock	0	4	4	5	87

6.6.2 Mood Classification

Table 6.12: Arousal classification performance measure

Classifier	Precision	Recall	F-Measure	Accuracy
SVM	0.70	0.74	0.72	70.69
ANN	0.75	0.72	0.72	73.10

For our final model we used ANN with these feature: Spectral centroid, MFCC, Zero

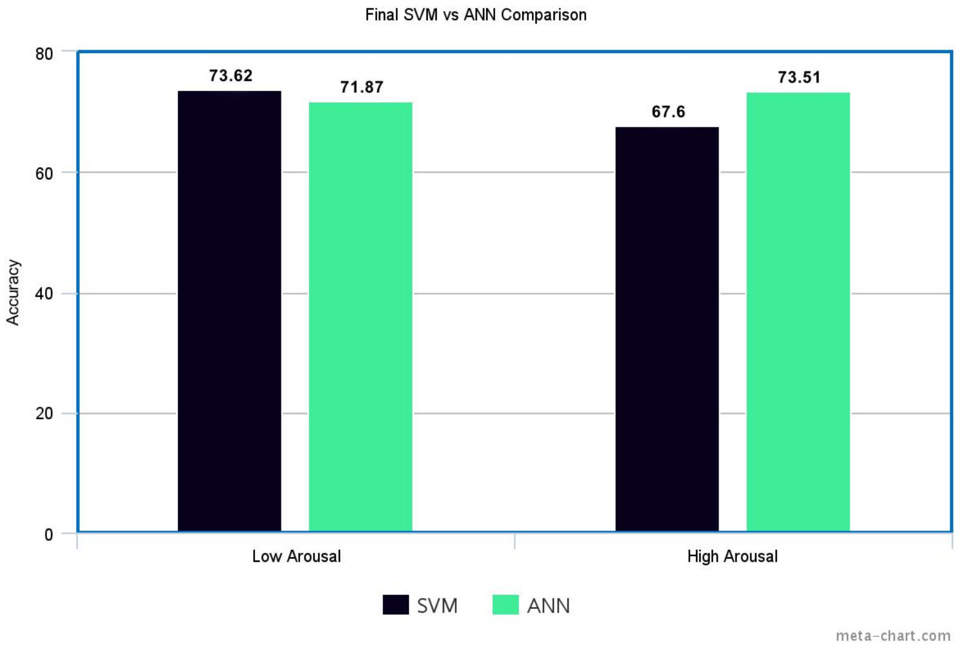


Figure 6.9: Final SVM and ANN comparision based on arousal

Crossing, Compactness, Rhythm, Spectral Flux, RMS and Spectral Variability.

Table 6.13: Arousal classification confusion matrix

	Low Arousal	High Arousal
Low Arousal	105	40
High Arousal	38	107

Table 6.14: Valence performance measure

Classifier	Precision	Recall	F-Measure	Accuracy
SVM	0.68	0.69	0.68	67.59
ANN	0.68	0.64	0.65	67.24

For our final model we used ANN with these feature: Spectral centroid, MFCC, Zero Crossing, Compactness, Rhythm, Spectral Flux, RMS and Spectral Variability.

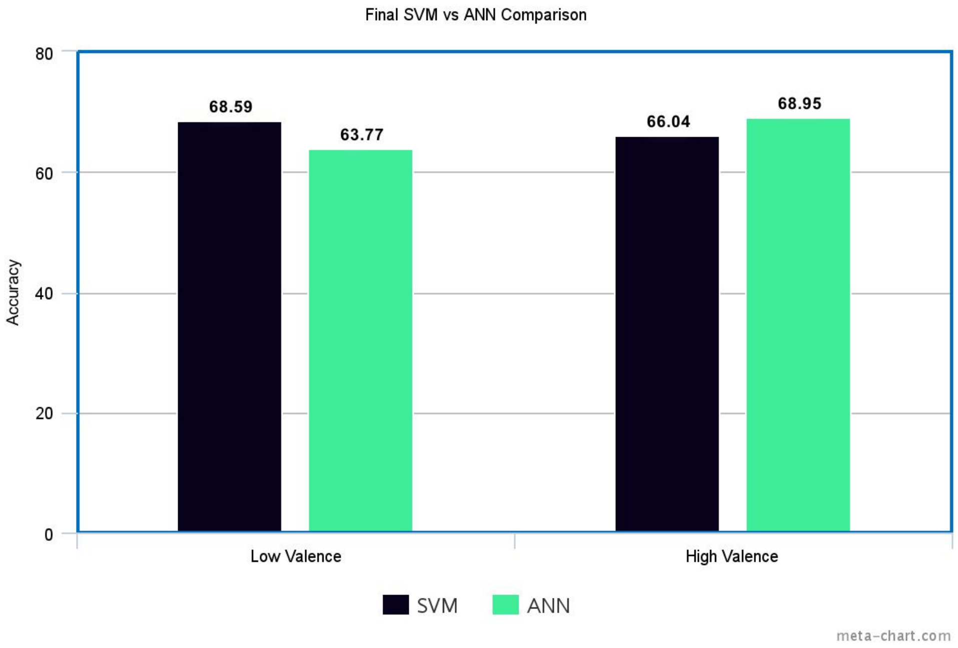


Figure 6.10: Final SVM and ANN comparision based on valence

Table 6.15: Valence confusion matrix

	Low Valence	High Valence
Low Valence	95	52
High Valence	43	100

6.7 Problems and Solutions

6.7.1 Solved Problems

(i) **Mp3 Decoding:**The Mp3 file format uses various lossy compression techniques that allows it to store audio data in a compact and efficient manner. That however means that its headers and file formats are pretty complicated and decoding them would require a substantial amount of work. So, we had problems reading and decoding the Mp3 file format.

We solved it by using the Mp3SPI library which read and gave us the samples of the mp3 files.

(ii) **Feature Integration:**We had trouble deciding ways in which we could integrate the different song features into a single feature-set that could represent the song. This was complicated by the fact that the features all varied in length, range as well as magnitudes. In the end, we concatenated the features into a single feature vector. However, by storing their means and standard deviation and normalizing them before usage, we minimized the effects of one feature over the others.

(iii) **Feature and Model Persistence:**We needed a way to store the extracted features and also the trained models. Doing so in a simple file or creating our own writer/parser for various appropriate formats would be error-prone and take up a lot of our time. So, we used GSON and XStream to store and also read them as JSON and XML files respectively.

(iv) **Pitch Extraction:**Pitch is the perceived fundamental frequency and hence it is subjective. So, there is no definite mathematical formulae or methods to represent or calculate it. So, we used the TarsosDSP library to extract pitch.

6.7.2 Unsolved Problems

(i) **Distance Measure for Songs:** One way to achieve song clustering or even classification is to develop distance measures to figure out the ”distance” or difference between any two given songs. So, we tried to do the same. However, our initial attempts at using a simple Euclidean Distance measure didnt prove successful and later attempts using Gaussian Mixture Models proved to be too computationally intensive to be useful.

(ii) **Heap Error on Long Songs:**Due to lack of memory optimization we run into a Heap error when processing songs longer than about six minutes. This might be solved by better profiling of the memory used or by using shorter sections of songs for classification.

7 CONCLUSION

7.1 Conclusion

Automatic Music classification is an interesting problem that has gained prominence with the rise of the Internet and the subsequent growth in the amount of music available to the general populace.

Any type of classification of music is difficult simply because the classifications themselves dont have a clear definition. Still, we can work with fuzzy boundaries between these classes to get good enough results with Music Classification Systems.

These systems broadly consist of two steps: Feature Extraction and Classification. However, they can be created by combining a variety of techniques and components.

So, in this project, many such components and approaches were studied such as: types and combinations of features for proper representation of songs, feature integration approaches, classifier types, and their parameters, etc.

All these studies were done in order to tackle two related but distinct problems:

- In Automatic Music Genre Classification (AMGC), very good performances were achieved with both of the classifiers employed: the final SVM model got 83 per cent accuracy and the ANN model getting 88 per cent accuracy for five genres. These results are comparable with the state-of-the-art results, especially involving the same dataset.
- In Music Mood Classification however, the good results couldnt be replicated. The result along both axes of the music mood model used (arousal and valence) were underwhelming. Around 73 per cent accuracy was achieved using ANN for the binary low/high arousal classification. SVM did even worse with around 70 per cent accuracy. For low/high valence classification, both of the classifiers settled on 67 per cent

accuracy.

7.2 Future Enhancements

There are many ways in which the project could be expanded.

- Adding many more genres would be a natural enhancement as the five genres used here dont even come close to encompassing all the variety of musical genres.
- The mood model could also be made a continuum in the two-dimensional space instead of hard binary classifiers.
- More features could be added to the feature set and studies conducted as to which combinations of features work best in which problem domain.
- Smart Playlist Generation System could also be built using these types of Automatic Music Classification Systems.
- The training set used to build the classifier could be increased in both size and variety to make the models generated more robust. This could also be a topic of study.

Bibliography

[1] Smith, Steven W. *The scientists and engineers guide to digital signal processing.* California Technical Publishing, 2013. Print.

[2] Kassler, Michael. *Toward musical information retrieval.* Perspectives of New Music (1966): 59-67.

[3] Andel, James. *On the segmentation and analysis of continuous musical sound by digital computer.* Diss. Stanford University, 1975.

[4] Tzanetakis, George, and Perry Cook. *Manipulation, analysis and retrieval systems for audio signals.* Princeton, NJ, USA: Princeton University, 2002.

[5] Alghoniemy, Masoud, and Ahmed H. Tewfik. *Rhythm and periodicity detection in polyphonic music.* Multimedia Signal Processing, 1999 IEEE 3rd Workshop on. Ieee, 1999.

[6] Byrd, Donald, and Tim Crawford. *Problems of music information retrieval in the real world.* Information processing and management 38.2 (2002): 249-272.

[7] Saunders, John. *Real-time discrimination of broadcast speech/music.* icassp. Vol. 96. 1996.

[8] Wold, Erling, et al. *Content-based classification, search, and retrieval of audio.* IEEE multimedia 3.3 (1996): 27-36.

[9] Scheirer, Eric, and Malcolm Slaney. *Construction and evaluation of a robust multifeature speech/music discriminator.* Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on. Vol. 2. IEEE, 1997.

[10] Casey, Michael A., et al. *Content-based music information retrieval: Current directions and future challenges.* Proceedings of the IEEE 96.4 (2008): 668-696.

[11] Kaminskas, Marius, and Francesco Ricci. *Contextual music information retrieval and recommendation: State of the art and challenges.* Computer Science Review 6.2 (2012): 89-119.

[12] Tzanetakis, George, and Perry Cook, Musical genre classification based on audio signals *IEEE Transactions on Speech and Audio Processing*, 10.5 (2002)

[13] Scaringella, Nicolas, Giorgio Zoia, and Daniel Mlynek. *Automatic genre classification of music content: a survey.* IEEE Signal Processing Magazine 23.2 (2006): 133-141.

[14] Perrot, David, and Robert Gjerdigen. *Scanning the dial: An exploration of factors in the identification of musical style.* Proceedings of the 1999 Society for Music Perception and Cognition. 1999.

[15] Lippens, Stefaan, Jean-Pierre Martens, and Tom De Mulder. *A comparison of human and automatic musical genre classification.* Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on. Vol. 4. IEEE, 2004.

[16] Kour, Gursimran, and Neha Mehan. *Music Genre Classification using MFCC, SVM and BPNN.* International Journal of Computer Applications (09758887) Volume (2015).

[17] Logan, Beth. *Mel Frequency Cepstral Coefficients for Music Modeling.* ISMIR. 2000.

[18] Li, Tao, Mitsunori Ogihara, and Qi Li. *A comparative study on content-based music genre classification.* Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. ACM, 2003.

[19] Anglade, Amlie, et al. *Improving music genre classification using automatically induced harmony rules.* Journal of New Music Research 39.4 (2010): 349-361.

[20] Muller, Mathias, et al. *Signal processing for music analysis.*, Selected Topics in Signal Processing, IEEE Journal of 5.6 (2011): 1088-1110.

[21] Dooling, Robert J and Stewart H Hulse. *The Comparative Psychology Of Audition.* Hillsdale, N.J.: L. Erlbaum Associates, 1989. Print.

[22] Prasad, Bhanu, and SR Mahadeva Prasanna, eds. *Speech, audio, image and biomedical signal processing using neural networks.* Vol. 83. Springer, 2007.

[23] Neumayer, Robert. *Musical genre classification.* (2004).

[24] Koerich, Alessandro L. *Improving the Reliability of Music Genre Classification using Rejection and Verification.* ISMIR. 2013.

[25] Haggblade, Michael, Yang Hong and Kenny Kao. Music genere classification. *Department of Computer Science*, Stanford University, 2011.

[26] Nasridinov, Aziz, and Young-Ho Park. *A Study on Music Genre Recognition and Classification Techniques.* International Journal of Multimedia and Ubiquitous Engineering 9.4 (2014): 31-42.

[27] Li, Tao, and George Tzanetakis. *Factors in automatic musical genre classification of audio signals.* Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.. IEEE, 2003.

[28] Downie, J. Stephen. *Toward the scientific evaluation of music information retrieval systems.* ISMIR. 2003.

[29] Tzanetakis, George, and Perry Cook, Musical genre classification *Journal of Personality*, 60(2):225251, 1992.

[30] Ekman, Paul. *Are there basic emotions?.* (1992): 550.

[31] James A. Russell. *A circumplex model of affect.* Journal of Personality and Social Psychology, 39(6): 1161-1178, 1980.

[32] Thayer, Robert E. *The biopsychology of mood and arousal.* Oxford University Press, 1990.

[33] Juslin, Patrik N., and John A. Sloboda. *Music and emotion: Theory and research.* Oxford University Press, 2001.

[34] Lu, Lie, Dan Liu, and Hong-Jiang Zhang. *Automatic mood detection and tracking of music audio signals.* IEEE Transactions on audio, speech, and language processing 14.1 (2006): 5-18.

[35] Fu, Zhouyu, et al. *A survey of audio-based music classification and annotation.* IEEE Transactions on Multimedia 13.2 (2011): 303-319.

[36] Soleymani, Mohammad, et al. *1000 songs for emotional analysis of music.* Proceedings of the 2nd ACM international workshop on Crowdsourcing for multimedia. ACM, 2013.

[37] Apon, Amy, et al. *Inital Starting Point Analysis for K-Means Clustering: A Case Study.* (2006).

[38] Jain, Anil K. *Data clustering: 50 years beyond K-means.* Pattern recognition letters 31.8 (2010): 651-666.

[39] Hamerly, Greg, and Charles Elkan. *Alternatives to the k-means algorithm that find better clusterings.* In Proceedings of the eleventh international conference on Information and knowledge management, pp. 600-607. ACM, 2002.

APPENDIX A

8.3 Signal Processsing

In signal processing, a window function (also known as an apodization function or tapering function) is a mathematical function that is zero-valued outside of some chosen interval. For instance, a function that is constant inside the interval and zero elsewhere is called a rectangular window, which describes the shape of its graphical representation. When another function or waveform/data-sequence is multiplied by a window function, the product is also zero-valued outside the interval: all that is left is the part where they overlap, the ”view through the window”.

Applications of window functions include spectral analysis, filter design, and beam forming. In typical applications, the window functions used are non-negative smooth ”bell-shaped” curves, though rectangle, triangle, and other functions can be used. A more general definition of window functions does not require them to be identically zero outside an interval, as long as the product of the window multiplied by its argument is square integral, and, more specifically, that the function goes sufficiently rapidly toward zero. One of the major applications of window functions includes the design of finite impulse response filters and the spectral analysis.

8.4 Spectral Analysis

The Fourier transform of the function cos t is zero, except at frequency . However, many other functions and waveforms do not have convenient closed form transforms. Alternatively, one might be interested in their spectral content only during a certain time period. In either case, the Fourier transform (or something similar) can be applied on one or more finite intervals of the waveform. In general, the transform is applied to the product of the waveform and a window function. Any window (including rectangular) affects the spectral estimate computed by this method.

8.5 Windowing

Windowing of a simple waveform like cos t causes its Fourier transform to develop non- zero values (commonly called spectral leakage) at frequencies other than . The leakage tends to be worst (highest) near and least at frequencies farthest from . If the waveform under analysis comprises two sinusoids of different frequencies, leakage can interfere with the ability to distinguish them spectrally. If their frequencies are dissimilar and one component is weaker, then leakage from the larger component can obscure the weaker ones presence. But if the frequencies are similar, leakage can render them irresolvable even when the sinusoids are of equal strength.

The rectangular window has excellent resolution characteristics for sinusoids of comparable strength, but it is a poor choice for sinusoids of disparate amplitudes. This characteristic is sometimes described as low-dynamic-range.

At the other extreme of dynamic range are the windows with the poorest resolution. These high-dynamic-range low-resolution windows are also poorest in terms of sensitivity; this is, if the input waveform contains random noise close to the frequency of a sinusoid, the response to noise, compared to the sinusoid, will be higher than with a higher-resolution window. In other words, the ability to find weak sinusoids amidst the noise is diminished by a high- dynamic-range window. High-dynamic-range windows are probably most often justified in wideband applications, where the spectrum being analyzed is expected to contain many different components of various amplitudes.

In between the extremes are moderate windows, such as Hamming and Hann. They are commonly used in narrowband applications, such as the spectrum of a telephone channel. In summary, spectral analysis involves a tradeoff between resolving comparable strength components with similar frequencies and resolving disparate strength components with dissimilar frequencies. That tradeoff occurs when the window function is chosen.

8.6 Filter Bank

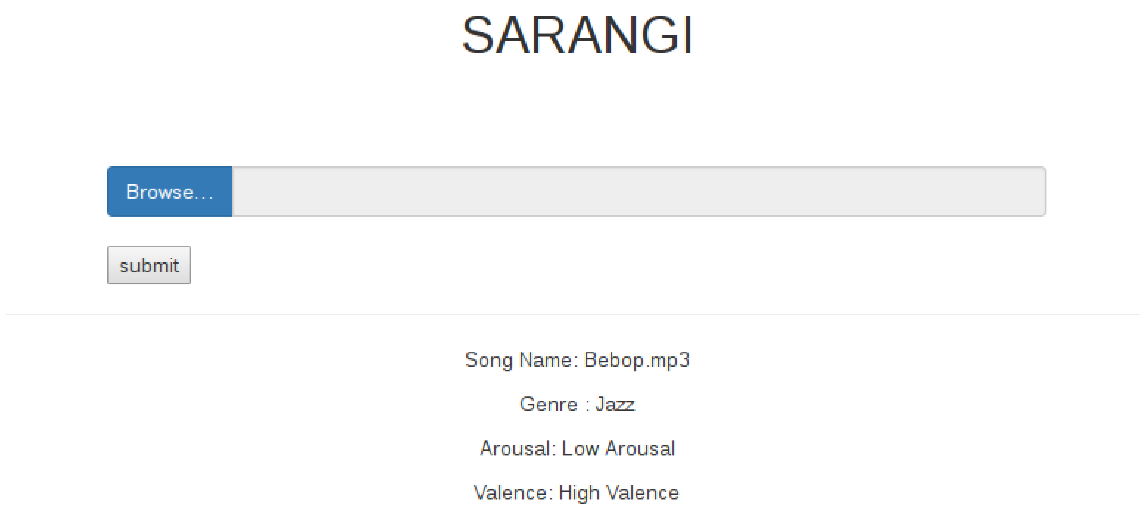
In signal processing, a filter bank is an array of band-pass filters that separates the input signal into multiple components, each one carrying a single frequency sub-band of the original signal. One application of a filter bank is a graphic equalizer, which can attenuate the components differently and recombine them into a modified version of the original signal. The process of decomposition performed by the filter bank is called analysis (meaning analysis of the signal in terms of its components in each sub-band); the output of analysis is referred to as a sub band signal with as many sub bands as there are filters in the filter bank. The reconstruction process is called synthesis, meaning reconstitution of a complete signal resulting from the filtering process.

In digital signal processing, the term filter bank is also commonly applied to a bank of receivers. The difference is that receivers also down-convert the sub bands to a low center frequency that can be re-sampled at a reduced rate. The same result can sometimes be achieved by under sampling the band pass sub bands.

Another application of filter banks is signal compression, when some frequencies are more important than others. After decomposition, the important frequencies can be coded with a fine resolution. Small differences at these frequencies are significant and a coding scheme that preserves these differences must be used. On the other hand, less important frequencies do not have to be exact. A coarser coding scheme can be used, even though some of the finer (but less important) details will be lost in the coding.

The vocoder uses a filter bank to determine the amplitude information of the sub bands of a modulator signal (such as a voice) and uses them to control the amplitude of the sub bands of a carrier signal (such as the output of a guitar or synthesizer), thus imposing the dynamic characteristics of the modulator on the carrier.

8.7 GUI



Song Name: Bebop.mp3

Genre : Jazz

Arousal: Low Arousal

Valence: High Valence

Figure 8.1: Graphics user interface