

**CLASSIFICATION OF SONGS BASED ON GENRE  
USING ML/DL ALGORITHMS**

Project report submitted in partial fulfilment of the  
requirement for the degree of Bachelor of Technology

in

**Computer Science and Engineering**

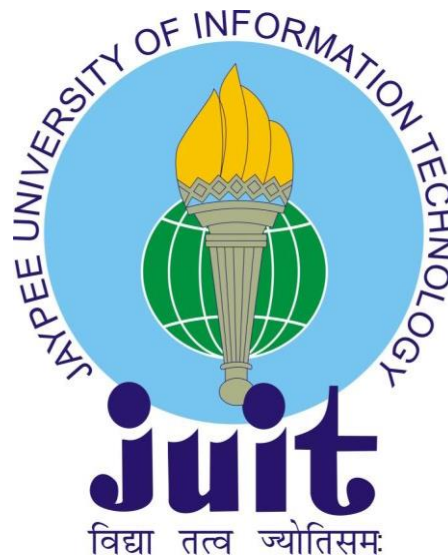
By

Ayush Dharmani (191262)

Under the supervision of

Mr. Prateek

to



Department of Computer Science & Engineering and  
Information Technology

**Jaypee University of Information Technology**

**Waknaghat, Solan**

# Certificate

## Candidate's Declaration

I hereby declare that the work presented in this report entitled “ **Classification Of Songs Based On Genre Using ML/DI Algorithms**” in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Wagnaghat is an authentic record of my own work carried out over a period from August 2022 to May 2023 under the supervision of **Mr. Prateek** (Assistant Professor, Department of CSE, Jaypee University of Information Technology, Wagnaghat).

I also authenticate that I have carried out the above mentioned project work under the proficiency stream **Machine Learning**.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature)

Ayush Dharmani

191262

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature)

Mr. Prateek

Assistant Professor (Grade II)

Computer Science & Engineering

Dated: May 1, 2023

**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT**  
**PLAGIARISM VERIFICATION REPORT**

Date: .....

Type of Document (Tick):  PhD Thesis  M.Tech Dissertation/ Report  B.Tech Project Report  Paper

Name: \_\_\_\_\_ Department: \_\_\_\_\_ Enrolment No \_\_\_\_\_

Contact No. \_\_\_\_\_ E-mail. \_\_\_\_\_

Name of the Supervisor: \_\_\_\_\_

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): \_\_\_\_\_

\_\_\_\_\_

**UNDERTAKING**

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/ revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

**Complete Thesis/Report Pages Detail:**

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

**(Signature of Student)**

**FOR DEPARTMENT USE**

We have checked the thesis/report as per norms and found **Similarity Index** at .....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

**(Signature of Guide/Supervisor)**

**Signature of HOD**

**FOR LRC USE**

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	<ul style="list-style-type: none"> <li>• All Preliminary Pages</li> <li>• Bibliography/Images/Quotes</li> <li>• 14 Words String</li> </ul>		Word Counts	
<b>Report Generated on</b>			Character Counts	
		<b>Submission ID</b>	Total Pages Scanned	
			File Size	

Checked by  
Name & Signature

Librarian

.....

**Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at [plagcheck.juit@gmail.com](mailto:plagcheck.juit@gmail.com)**

## Acknowledgement

All compliments and praise are due to God who empowered me with strength and sense of devotion to successfully accomplish this project work successfully.

I am really grateful and wish my profound indebtedness to Supervisor **Mr. Prateek, Assistant Professor (Grade II)**, Department of CSE Jaypee University of Information Technology, Wagnaghat. Deep Knowledge & keen interest of my supervisor in the field of “**Machine Learning**” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to **Mr. Prateek**, Department of CSE, for his kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Ayush Dharmani  
191262

## Table of Contents

<b>Chapter Number</b>	<b>Title</b>	<b>Page Number</b>
1.	Introduction	01-09
1.1	Introduction	03-05
1.2	Problem Statement	05-06
1.3	Objectives	06-07
1.4	Methodology	07-08
1.5	Organisation	08-09
2	Literature Review	10-26
3	System Design & Development	27-39
3.1	Algorithm Analysis	27
3.2	Designing of Algorithm	27-29
3.3	Model Development	29-39
4	Experiments & Result Analysis	40-51
4.1	Requirements	40-42
4.1.1	Language Used	40
4.1.2	Libraries Used	40-42
4.1.3	System Requirements	42
4.1.4	Hardware Requirements	42
4.2	Result at various stages	42-51
5	Conclusions	52-55
5.1	Conclusions	52-53
5.2	Future Work	53-55
5.3	Applications of the Project	55

6	References	56-57
7	Appendices	58-63

## List of Abbreviation

<b>S.No.</b>	<b>Abbr.</b>	<b>Full Form</b>
1	AI	Artificial Intelligence
2	CNN	Convolutional Neural Network
3	SVM	Support Vector Machine
4	MIR	Music Information Retrieval
5	ML	Machine Learning
6	MFCC	Mel Frequency Cepstral Coefficients
7	KNN	K-Nearest Neighbor
8	GMM	Gaussian Mixture Models
9	LDA	Linear Discriminant Analysis
10	DWCH	Daubechies wavelet

## List of Figures

<b>Figure Number</b>	<b>Title of Figure</b>	<b>Page Number</b>
1	Support Vector Machine	02
2	Base Architecture of CNN Model	03
3	Artificial Intelligence vs Machine Learning vs Deep Learning	10
4	Supervised Learning vs Unsupervised Learning	11
5	Architecture of Neural Network	12
6	CNN Model used by Authors	16
7	Taxonomy Structure for Dataset A	25
8	Taxonomy Structure for Dataset B	26
9	One vs Rest method of SVM	28
10	CNN Model	29
11	Files inside dataset	30
12	Audio File available on colab	34
13	Raw Wave plot of audio file	34
14	Spectral Rolloff of audio file	35
15	Spectrogram of audio file	35



16	Chroma Features of audio file	36
17	Zero Crossing Rate of audio file	36
18	Final Flow Diagram of our CNN Model	39
19	First 5 rows of dataset	42
20	Columns available in the dataset	43
21	Labels/Genres in Dataset	43
22	Data Type of audio after loading	44
23	Features extracted after loading in colab using Librosa	44
24	Wave plot for Pop song	44
25	Spectrogram for Pop song	45
26	Spectral Rolloff of uploaded Pop song	45
27	Chroma Features plot of Pop song	46
28	Zero Crossing Rate Plot of Pop song	46
29	SVM Model scores	47
30	Running 600 Epochs	48
31	Epoch training of model	48
32	Accuracy of CNN Model	49
33	Accurate prediction on sample data	49

34	A reggae genre song file uploaded	49
35	Reggae genre song correctly identified as “Reggae”	50
36	Another Rock genre song correctly identified as “Rock”	50

## List of Graphs

<b>Table Number</b>	<b>Title of Table</b>	<b>Page Number</b>
1	Gaussian Kernel Graph	32
2	Sigmoid Kernel Function	32
3	Polynomial Kernel Function	33
4	RELU Activation Function	38

## List of Tables

<b>Table Number</b>	<b>Title of Table</b>	<b>Page Number</b>
1	Number of Audio Clips in each genre	05
2	Result analysis of Authors CNN model	17
3	Comparison of accuracy of SVM and CNN	51

## **ABSTRACT**

The music industry has undergone significant changes in recent years, both in its conventional existence and in the form of music created. The popularity of the music industry has grown, and with it, the market for different music styles. Music not only brings individuals together, but also provides insight into various cultures. To deliver better recommendations and suggestions to people, music needs to be classified into genres to meet the categorical needs of consumers. However, classifying music into different genres is a challenging task in the area of music information retrieval (MIR), and there have been several attempts to classify music using various machine learning approaches.

The primary objective of this project is to automatically classify audio files into their respective musical genres. To achieve this goal, we will compare the performance of two classes of models: Support Vector Machines (SVMs) and Convolutional Neural Networks (CNNs). SVMs and CNNs are two commonly used approaches for classification under machine learning and deep learning, respectively, and have shown promise in delivering effective regression and classification results. The CNN model we will use is trained end-to-end to predict the genre label of an audio signal using its spectrograms, spectral roll-off, chroma features, and zero-crossing rate. We will use a dataset of audio tracks with similar sizes and frequency ranges, specifically the GTZAN genre classification dataset. The dataset contains around 1000 audio files, each with a duration of 30 seconds, and 10 music genres (10 classes), with each class containing 100 audio files/tracks. Each track is in .wav format.

In summary, this project aims to classify audio files into different genres using two models, SVMs and CNNs, and comparing their performance. The chosen dataset is the GTZAN genre classification dataset, and the low-level features of frequency and time domain will be used to classify the audio files.

## **Chapter 1: INTRODUCTION**

In the music industry and in the field of music information retrieval, the classification of songs based on their genre is a crucial task. Users may quickly choose and listen to their favourite kind of music, and it helps business experts create targeted marketing campaigns for different genres. In order to address this classification issue, our study concentrated on creating an autonomous song genre classifier model utilising machine learning and deep learning methods.

The dataset which was chosen for recommendation in the project included songs from various genres, and utilised a content-based approach rather than collaborative filtering to classify the songs based on their audio features. This approach allowed to classify songs based solely on their intrinsic characteristics, without relying on external information such as user preferences. To achieve this, we applied a range of machine learning and deep learning techniques, including convolutional neural networks (CNNs), which are effective at handling audio and image data.

The process involved several steps, including data pre-processing, feature extraction, and model development. During data pre-processing, audio feature extraction was performed using the Mel-Frequency Cepstral Coefficients (MFCCs) technique to extract audio features from the audio signals of the songs. Then applied various machine learning and deep learning models, including CNNs, to classify the songs into the appropriate genres.

The aim of this project is to develop a reliable and accurate model that could effectively classify songs into the appropriate genres, benefiting both users and industry professionals in the music domain. The resulting model is expected to be useful for users in easily finding and listening to their preferred music genre and for industry professionals in developing targeted marketing strategies for different genres.

**Support Vector Machine (SVM):** One of the most popular supervised learning algorithms, Support Vector Machine, or SVM, is utilised for both classification and regression tasks. However, the majority of the time it is utilised for machine learning classification issues. The basic goal of the Support Vector Machine approach is to establish a decision boundary or best line fit that can categorise the n-dimensional space, allowing us to forecast a new data point's class simply by placing it in the appropriate category. These are termed support vectors because they provide support for the hyperplane. Support vector machines may be divided into two basic categories: linear SVMs and non-linear SVMs.

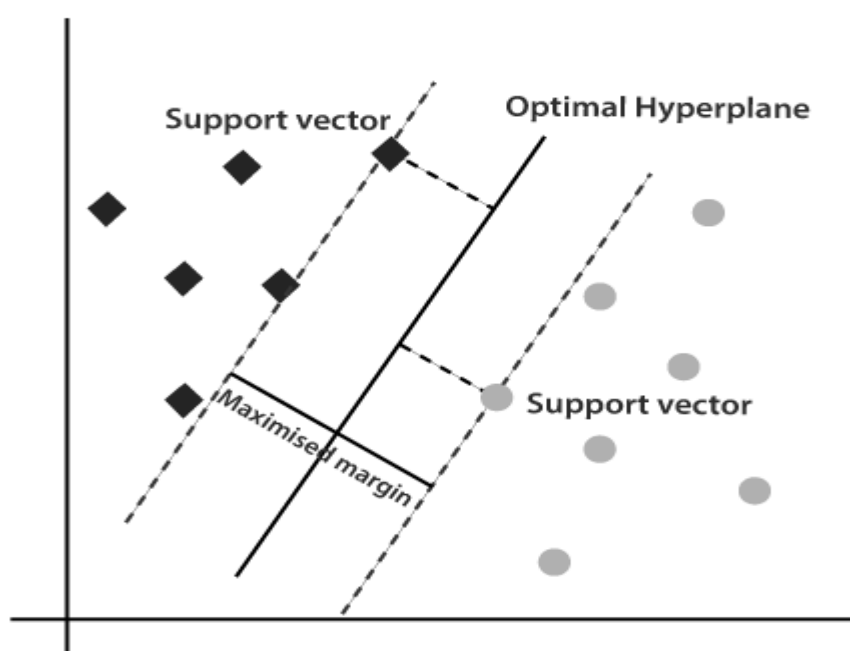


Figure 1: Support Vector Machine

**Convolutional Neural Networks (CNN):** Traditional, fully linked multilayer perceptron models do reasonably well in image identification applications. Due to the limited computational power, they do not scale well with high quality photographs. Furthermore, because multilayer perceptrons do not account for the spatial organisation of visual patterns, distant pixels might have an equal

influence on the identification of an area as a nearby pixel. CNNs work around this issue by using 3D layers that are only linked to a tiny portion of the prior layer, and by sharing the weights and biases of filters within the same layer.

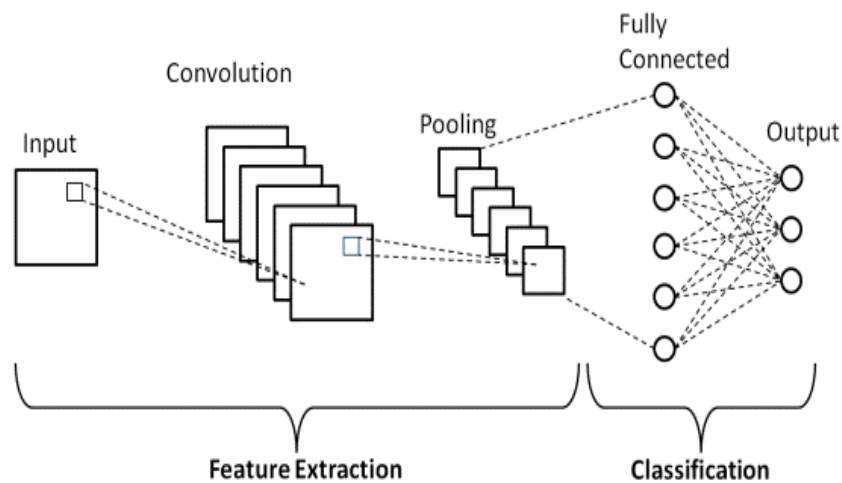


Figure 2: Basic Architecture of CNN model

The above figure represents the basic architecture for the CNN model. The audio input goes through various layers of convolutional layer network. Added a filter that convolves through the input and then the output layer generated acts as the input layer for the next one, this process goes on. The Max pooling layer further picks up the most prominent feature of the previous feature map.

### 1.1 Introduction

The development in the field of technology has led to the widespread adoption of machine learning solutions across multiple industries. The importance of categorising music by genre rests in its fundamental function in creating reliable recommendation systems for music organisations. Numerous machine and deep learning algorithms have been used throughout the years to conduct substantial study on the classification of musical genres. Machine learning algorithms have the potential to analyse large datasets automatically and identify interesting patterns. The field of genre classification and music information retrieval has



been the subject of extensive research, benefiting both the music industry and consumers. Digitalization has revolutionised the music industry, from the creation and production to the consumption and sharing of music.

Music is categorised into subjective categories called genres, and human experts have traditionally been relied upon to attribute genre tags to songs. With the ever-growing customer base, there has been an increase in demand for various music styles. Music not only brings people together, but it also provides insight into different cultures. Hence, it is essential to categorise music according to genres to satisfy the needs of people categorically.

A substantial amount of research has been done on classifying music genres, which may be divided into two types. The GTZAN and FMA databases are two that are often used. Deep learning has been an effective method for creating reliable end-to-end systems in a variety of image, video, audio, and voice analysis domains.

This study is subject to certain limitations that relate to the quality of the data or the sampling method used, as well as the restricted size of the dataset. Additionally, there may be challenges in terms of feature engineering or optimising the selected feature set. It's also vital to keep in mind that not all methods for the task at hand may have been included in this study.

The GTZAN music audio file collection, which is widely used by music information retrieval researchers, was used as the dataset for this study. Ten different music genres were used for classification, including Jazz, Rock, Pop, Country, Hip-Hop, Reggae, Classical, Rhythm and Blues, Metal, and Disco. The dataset consists of approximately 1,000 music pieces, each of 30 seconds, and categorised into 10 different musical genres.

Table 1: Number of Audio Clips in each genre

S.No.	Class	No. of Clips/Files
1.	Rock	100
2.	Reggae	100
3.	Pop	100
4.	Metal	100
5.	Jazz	100
6.	Hip-Hop	100
7.	Disco	100
8.	Country	100
9.	Classical	100
10.	Blues	100
	<b>Total</b>	<b>1000</b>

## 1.2 Problem Statement

The purpose of our research is to develop a classification model using deep learning and machine learning methods that accurately classifies songs and music into the available categories. This study investigates methods for managing sound files in Python, extracting pertinent sound and audio properties, and using machine learning and deep learning algorithms to determine the genre of an audio file.

In order to do this, a variety of machine learning methods, such as k-nearest neighbour (kNN), k-means, multi-class SVM, and neural networks, have been examined and evaluated for their efficacy in categorising four genres, namely classical, jazz, metal, and pop. The pictures of the songs are also clustered based on their genres and characteristics are extracted using the Fourier-Mellin 2D transform.

Through a comparison of traditional machine learning techniques, the study intends to build a genre categorization model. This comparison analysis goal is to determine which algorithm will be best for creating the genre categorization model.

The evaluation of the algorithms will be based on the accuracy of the classification model and the algorithm with the highest accuracy will be selected as the optimal solution for building the genre classification model.

The results of this project can be used to improve the accuracy and efficiency of music classification models, which can have practical applications in music recommendation systems, audio search engines, and other related areas.

In summary, the main aim of this project is to investigate the feasibility and effectiveness of machine learning and deep learning algorithms for classifying music samples into different genres, with a view to building a robust and accurate genre classification model.

### **1.3 Objective**

The objective of our project is to create a machine learning/deep learning model that can categorise audio recordings into the appropriate musical genres automatically. In this study, we will examine how well Support Vector Machine (SVM) and Convolutional Neural Networks (CNN) perform in terms of producing accurate results for classification. To predict the genre label of an audio signal, the CNN model is trained end-to-end utilising spectrograms, spectral roll-off, chroma characteristics, and zero crossing rate. The audio files are categorised using low-level frequency and temporal domain characteristics.

Mel Frequency Cepstral Coefficient, spectrograms, raw audio and Musical Instrument Digital Interface (MIDI) files are some of the input formats that have

been employed for learning purposes. The major focus of this study is on the many kinds of produced spectrogram pictures that may be applied to neural network training. These spectrograms describe a song's spectrum information in the visual domain and this study takes into account three different spectrogram types: linear, Mel frequency logarithmic, and spectrograms.

The auditory range, which people can detect more clearly, is given greater weight in the spectrogram that has been Mel scaled. The human ear is more sensitive to differences between lower frequencies than it is to differences between higher frequencies. This is taken into account in the Mel scaled spectrogram by emphasising the lower frequencies on the Y axis rather than the higher ones. Equally on the Y-axis are the linear spectrograms.

Overall, the project aims to develop a machine learning/deep learning model that efficiently classifies songs/music based on different genres. The performance of SVM and CNN models will be compared, and the classification will be based on spectrograms, spectral roll-off, chroma features, and zero crossing rate. The classification model will enable users to quickly and easily select songs based on their genre, thereby reducing the time and effort required for manual classification.

#### **1.4 Methodology**

This section describes our methodology, which involves training a CNN architecture on song spectrograms that have already been classified with their musical genre. The suggested CNN advances the study by addressing flaws in the existing work, expanding it by incorporating new types of spectrograms, and extending it by doing an analysis of the spectrogram data. The spectrograms of songs and music that have not yet been processed by the CNN, or the testing set, can be used to classify the genre of music using the CNN model that has been created as a consequence of the training (by dataset). The basic methodology that we will follow during our study is as follows:

1. To find a perfect dataset that includes all required features.
2. The pre-processing of dataset and visualization of the spectrograms from the dataset for the CNN.
3. Feature scaling and cleaning of dataset.
4. Building a SVM Model.
5. Building a CNN architecture/model.
6. Testing of the trained model and checking for accuracy.

## **1.5 Organization**

This project report is divided into five distinct chapters, each of which is explained below:

**Chapter 1:** It includes a brief overview of our project, the problem statement that served as the foundation for our project's objective, our main project goals, and the methodology or approach that will be used to complete our study, which was to develop a classification model that accurately classifies songs into various genres.

**Chapter 2:** The history of music genre categorization is covered in this chapter using data from standardised books, journals, transactions, websites, and other sources. Machine learning, neural networks, and an introduction to a few distinct types of neural networks are the first topics covered. This chapter outlines how several neural network types, datasets, and accuracy evaluation approaches have been used by various academics to tackle music genre categorization model challenges, with variable degrees of success. It also explains the technique or strategy we decide to employ.

**Chapter 3:** This chapter's main objective is to describe in detail how to create the environment for the study that will be done for this study. Finding a dataset of songs or musical works to serve as a starting point, followed by the creation and visualisation of spectrograms from the musical works, and ultimately the

pre-processing of the dataset to train the model. The neural network that was used and the factors, including the software and hardware, that will be used to build the mode are mostly covered. Finally, it provides a list of all the various accuracy measures used in this study to assess the correctness of our model.

**Chapter 4:** This chapter provides the tests performed and the corresponding data gathered at various stages of the project to give us a clear picture of the accuracy of the model generated. The model that best matches the data is created once an experiment is complete by building on each stage. The accuracy of the various audio input types was assessed in this chapter before wrongly labelled network outputs were examined to see if the input had been appropriately classified.

**Chapter 5:** The study reported in this report is summarised in this chapter, together with the results of our end project model's outputs. It also identifies areas where it might be strengthened or expanded upon through more study and future development. It also includes the creative labour, inventions, and fresh ideas that came from the examination of the work and the results.

## Chapter 2: LITERATURE REVIEW

### Machine Learning

Although the terms "machine learning," "deep learning," and "artificial intelligence" are sometimes used interchangeably, they refer to different subfields of the larger science of AI. Machine learning is a method for creating an algorithm that mimics human thinking and generates output depending on data input. It involves the study of data to automatically create a model. Instead of depending on fixed instructions and conditions as traditional programming does, machine learning includes the system learning from the provided data and adapting to changes. Machine learning has grown significantly in popularity in recent years as a result of its capacity to deliver accurate findings across a range of fields.

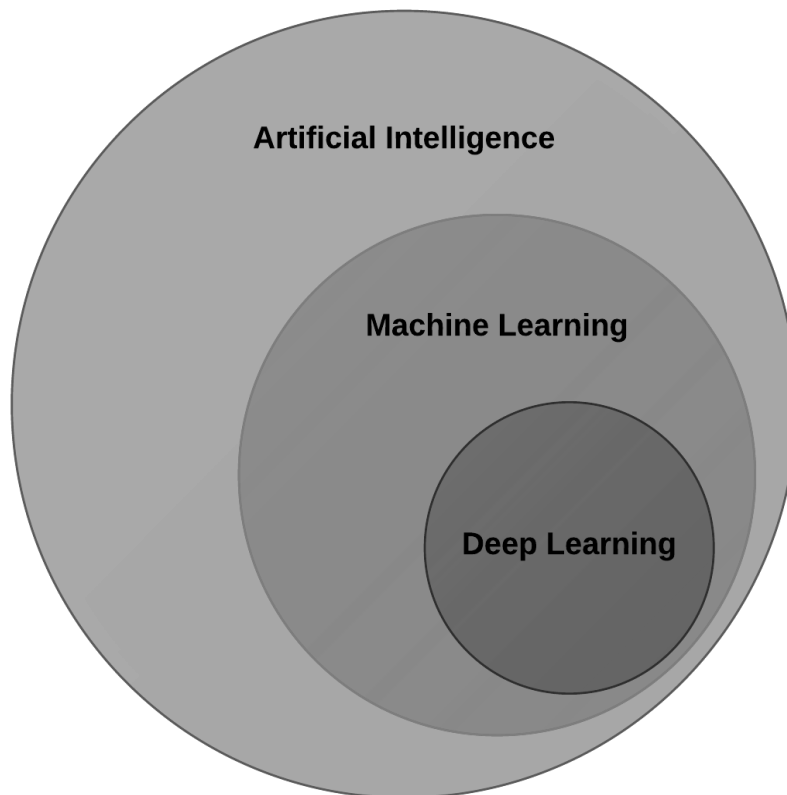


Figure 3: Artificial intelligence vs Machine Learning vs Deep Learning

A fully labelled dataset is used in supervised learning to create a mathematical model that forecasts outputs based on inputs. This method may be applied to the categorization of musical genres, where the model forecasts the musical genre based on the processing of incoming data. On the other hand, unsupervised learning entails taking valuable characteristics from unlabelled datasets without having a predetermined objective in mind. It is mostly used to group datasets together and find data trends.

By building a mathematical model based on the fully labelled dataset, supervised learning may be utilised to anticipate the type of music in the GTZAN dataset. Unsupervised learning, on the other hand, makes generalised efforts to extract relevant characteristics from the unlabelled dataset. This method can help uncover outliers and abnormalities by clustering the data and recognising trends.

Semi-supervised learning uses both labelled and unlabelled datasets to improve learning accuracy. This approach has been shown to significantly enhance the learning accuracy compared to unsupervised learning, particularly when dealing with limited labelled data. Reinforcement learning is different from the other learning algorithms, as it uses feedback mechanisms to learn. This approach involves rewarding correct actions and predictions to minimise risk and maximise reward in game environments.

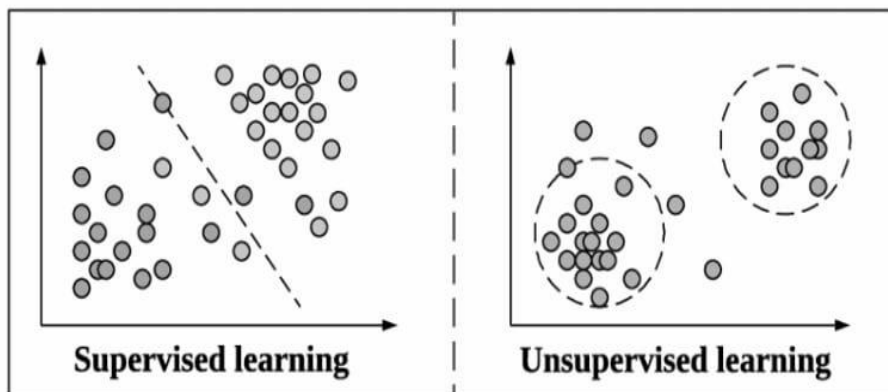


Figure 4: Supervised learning vs Unsupervised learning



## Neural Networks

The process of identifying significant features from large, complicated data sets and creating models or functions that reflect those characteristics is known as neural network (NN) training. NN training is a popular and efficient machine learning approach. In order to build a model, NNs normally need a training dataset. The model may then be used to categorise data using the characteristics discovered during training.

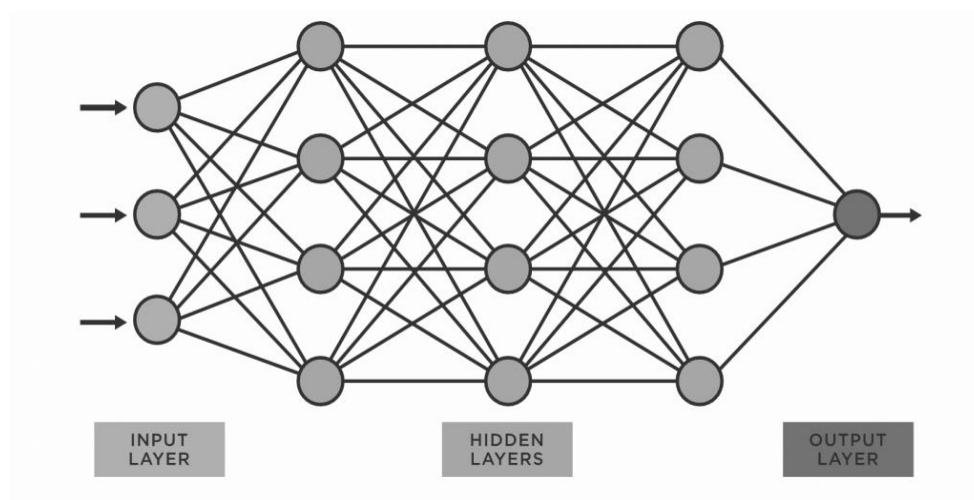


Figure 5: Architecture of Neural Network

Various papers were reviewed during our research and came across some interesting papers out of which one was “**Music genre classification using machine learning**” by Anirudh Ghildiyal, Komal Singh, Sachin Sharma. The music industry is expanding quickly thanks to new technical advancements in the modern world, where technology is developing and modernising at a breakneck rate. Therefore, the study of music and songs is becoming more and more interesting to scholars.

The author of this study report, which was published by IEEE, drew inspiration from a number of prior studies in which researchers trained their models using 3 second audio/song samples that were taken from the GTZAN dataset using a

residual neural network (RNN). A final accuracy of 90% was attained by the model, which additionally took into account the overlapping properties of other genres in the provided dataset. Similar to this, in another research the author referenced, several techniques and fixes were recommended, and a new near real-time classification using RNN was developed, although with a low accuracy of 64%. The researchers used the mean and covariance of Mel-frequency cepstral coefficients, or MFCC, to train their classification algorithm.

The MFCCs (Mel-frequency cepstral coefficients) and other characteristics of the songs that were collected from the GTZAN dataset were used to train the classification model, which resulted in a convolution neural network (CNN) that created a model with demonstrably high accuracy. Researchers also contrasted several pre-existing categorization models. The greatest machine learning/deep learning method for classifying music genres that may be used and create a model with definitely high accuracy was sought after by the author.

Convolution neural networks (CNN) were employed by some researchers to categorise the songs into several genres. There are three primary ways to visualise audio files: using spectrograms, chromagrams, and deriving Mel-frequency cepstral coefficients, or MFCC. primary authors employed various visualisation techniques.

In one of the research projects from which the author took inspiration, the researcher used the mel-spectrogram as an input to the model. The author used a duplicate convolution layer to enhance classification, after which the output was routed via numerous pooling layers and the network was subjected to statistical analysis.

When the author examined the Long Short-Term Memory (LSTM) model, the convolution neural networks (CNN) model, and the Mel-frequency cepstral coefficients (MFCC) model, the CNN model offered a superior accuracy. The

CNN featured five convolution layers, each with 32 nodes, the first of which was a fully linked layer with 128 nodes. The output layer, which contained 10 nodes, came after this one. There were five layers in the LSTM model, the first of which had 128 nodes and the following four each having 32. The CNN model has better accuracy and a higher rate of prediction.

In the given research paper, the authors, after analysing all the available and different algorithms for building a machine learning/deep learning classification model for genre classification of songs, the authors started to build a CNN model for genre classification. For achieving this, they followed the following steps:

1. Dataset Preparation:

The authors of this study used the GTZAN dataset for their analysis. The 10 categories in this dataset are Blues, Classical, Country, Disco, Hip-Hop, Jazz, Pop, Metal, Reggae, and Rock. 100 audio samples, each 30 seconds long, in.wav file, with 16-bit sampling at a frequency of 22050 Hz, are provided for each lesson.

2. Pre-Processing of Dataset:

With only around 100 audio clips per class, the dataset used for training the model may not be sufficient to achieve high accuracy. To address this, one approach is to increase the dataset size either by collecting more data or by using data augmentation techniques. Data augmentation can involve techniques such as changing the pitch or speed of the audio, adding background noise, or combining different audio samples. This can effectively increase the number of training and testing samples, helping the model to better learn the underlying patterns in the data and improve its accuracy and generalizability.

3. Spectrogram Generation:

The audio clips are fed into Colab using the librosa library in the first

stage, and each audio file in the dataset is transformed into its mel-spectrogram before being created. The spectrogram was then divided into 64 strips once it had been created. The data samples were multiplied by 64 as a result. For both the model's training and testing, a total of 64000 samples in each dimension of 480x10 were observed. The 64000 photos were then separated into 44200 for training, 7000 for validation, and 12800 for testing.

4. Feature Extraction:

Each and every audio file in the dataset may be represented as an audio signal, and each of these signals has a unique set of attributes. Thus, the audio elements that are pertinent to fixing the issue and helpful for model construction are removed. These definitions, which divide these traits into two subcategories, are motivated by the writings of the author:

a) Time Domain Feature:

i) Zero Crossing Rate:

It measures how quickly a signal shifts from positive to negative or the opposite.

ii) Root Mean Square Energy:

A signal's RMSE, or Root Mean Square Energy, can be interpreted as its volume. RMSE is also defined as:

$$\sqrt{\frac{1}{N} \sum |x(n)|^2} \text{-----Eq. 2.1}$$

b) Frequency Domain Features:

i) Mel-Frequency Cepstral Coefficient: These are the roughly

15-20 characteristics that make up the Mel-Frequency Cepstral Coefficient, which specifies the form of the audio signal.

ii) Chroma Features: The whole spectrum of a music signal is projected onto 12 bits, which stand for the 12 different semitones of an octave in music.

iii) Spectral Centroid: It is the weighted mean of the frequencies present in the sound that tells about the ‘center of mass’ of the signal.

iv) Spectral Roll-off: It is the value of frequency below which a specified percentage of the total spectral energy lies.

#### 5. CNN Model:

Two deep neural network sub-networks receive the spectrogram slices that served as the training images. In order to extract features from the input photographs, the authors' first neural network—a four-layer convolutional neural network—is used. The attributes that were obtained from this are given to the second sub-network, which categorises them. This network is totally connected and contains two tiers that are entirely interconnected. A substantial layer that is present at the very end foreshadows the genre of the transmitted audio.

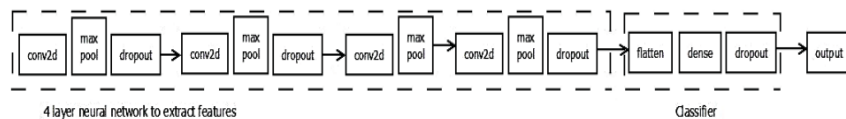


Figure 6: CNN Model used by Authors

#### 6. Results and Accuracy:

The CNN model used in the research achieved an accuracy of 91%,

indicating that it was successful in correctly classifying the input data 91% of the time. This level of accuracy suggests that the model is likely to be effective in its intended application and could potentially be useful in practical settings.

Table 2: Result analysis of Author’s CNN model

<b>Class</b>	<b>Sensitivity</b>	<b>Specificity</b>	<b>Accuracy</b>
Blues	0.93	0.93	0.93
Classical	0.92	0.98	0.95
Country	0.87	0.87	0.87
Disco	0.90	0.85	0.88
Hip-Hop	0.91	0.89	0.90
Jazz	0.91	0.93	0.92
Metal	0.93	0.97	0.95
Pop	0.91	0.90	0.90
Reggae	0.91	0.86	0.89
Rock	0.86	0.86	0.86
<b>Overall Accuracy</b>			<b>0.91</b>

Sensitivity, specificity, and accuracy are the three variables that are used to measure model performance in the table above. Here, the terms sensitivity and specificity describe how well the model categorises existing classes and non-classes, respectively. Information on the total percentage of properly identified occurrences is provided by accuracy. All of these criteria are described as:

- Sensitivity =  $TP / (TP + FP)$
- Specificity =  $TN / (TN + FP)$

- $\text{Accuracy} = (\text{TP} + \text{TN}) / \text{Total Events}$

Second paper which we reviewed was “**Music Genre Classification: A Review of Deep Learning and Traditional Machine-Learning Approaches**” by Ndiatenda Ndou, Ritesh Ajoodha, Ashwini Jadhav. The authors of this research investigated autonomous music genre classification with the aim to show that music can be classified merely on the audio signal itself utilising machine learning and deep learning approaches.

To carry out the research, three phases—“Phase A,” “Phase B,” and “Phase C”—were utilised. In phases “A” and “B,” they automated the classification of musical genres using six classical machine learning classifiers. However, input qualities of various dimensions are tested in the two phases. Phase “C” offers deep learning and machine learning approaches with more audio samples but a shorter length.

Their study made use of the GTZAN dataset. In one of their investigations, the original data set was duplicated and divided into 10,000 segments, each lasting three seconds. Despite the fact that this strategy produced more training data, the dataset's sample size per genre was variable, with certain genres having slightly fewer than or more than 1,000 songs.

Four characteristics that are frequently thought to help in the accurate classification of musical genres were used to assess their model. Important pre-processing experiments were conducted to prepare the raw data for the classification job before choosing these characteristics for model implementation. In this study, feature extraction was done for two reasons:

1. **Dimensionality reduction:** The raw data size is typically too large. Thus, the size of the entire raw audio file may prevent efficient processing. According to related research, feature sets may be used to

represent data with fewer values and provide a single feature value for the entire audio signal.

2. **Meaningful representation:** The information can all be extracted and used from a raw audio file, but it's important to convey musical aspects in a way that both computers and people can comprehend.

They obtained an n-dimensional vector after computing a song's characteristics. The length of the audio item under analysis determines the value of n. The following feature representations are investigated/explored for feature vectors  $V = (v_1, v_2, v_3, \dots, v_n)$  since large values of n require the usage of high-dimensional feature vectors, which are costly to analyse.

- Mean
- Standard Deviation
- The Feature Histograms
- MFCC Aggregation
- Area Moments

Feature Extraction was followed by Feature Selection. Reducing duplicate and unnecessary data is crucial for improving model learning accuracy and cutting training time. To assess the contribution of various characteristics to the accurate classification, an information gain ranking method was employed. The research done for this study was divided into three studies, or phases, and three separate feature sets were presented in each phase.

Support Vector Machines, Multilayer Perceptrons, Linear Logistic Regression, and K-Nearest Neighbours were only a few of the machine learning techniques employed by the authors. The author also used deep learning concepts to more precisely quantify metrics. The convolutional neural network (CNN) architecture used in this study was developed using Keras. The convolutional



neural network (CNN) architecture used in this study was developed using Keras. The input layer's five convolutional blocks and the CNN were created which included following steps:

- Convolutional layer with mirrored padding, 1x1 stride, and 3x3 filter
- The rectified linear activation function (ReLU)
- Maximum pooling with 2x2 stride and window size
- Probability of 0.2 for dropout regularization

The probabilities of the 10 label classes are produced by the CNN's final layer through fully linked layers that employ the SoftMax activation function.

Before the model identified the test data set, the author ran 3-fold repeated 10-fold cross-validation to eliminate bias. Analyse each model's performance in terms of classification accuracy and training duration.

It was found that, aside from the Multilayer Perceptron, the logistic regression had the longest training period for reaching the result reached in this work. Linear and logistic regression produced the greatest classification accuracy at 81%. All trained classifiers outperformed the Naive Bayes classifier as well.

The best classification accuracy was obtained by a support vector machine (SVM), which was 80.80%.

The best classification accuracy was attained by k-Nearest Neighbour (kNN), which also had the fastest training time (78 ms). The "Phase C" feature employed a 3 second feature set to 30 seconds.durations from phases A and B were used.

The classification accuracy produced by CNN is hence substantially lower than that provided by conventional machine learning models in the deep learning assessment. When employing a 3-second function set, CNN's best classification

accuracy is 72.40%. More training data is provided by the feature set with a 3 second length. The CNN implementation had the lowest accuracy (53.50%) for tasks lasting 30 seconds. However, owing to time and computational constraints, they were unable to expand the number of epochs beyond 120 in their investigation. It was discovered that the implementation of CNN utilising spectrograms enhanced the accuracy as the number of epochs rose.

Third paper which we reviewed was “**A Comparative Study on Content-Based Music Genre Classification**” by Tao Li, Mitsunori Ogihara, Qi Li. In this study, they put forth the DWCH feature extraction approach, which is based on wavelet coefficient histograms. By constructing histograms of Daubechies wavelet coefficients at various frequency subbands with various resolutions, DWCH conveys both local and global information. The accuracy of music genre classifications was greatly increased by combining DWCH with cutting-edge machine learning methods.

They also examined which of the features proposed worked best. Combining sound features with mel-frequency cepstrum coefficients, it was found that high accuracy is achieved with any of the tested multi-class classification algorithms.

To identify between combinations of sounds that could have the same or comparable rhythmic and pitch content, utilise the timbre text feature. These functions are used through speech recognition. The audio stream is initially divided into statistically stationary frames in order to extract sound characteristics. This is often accomplished by repeatedly using a window function. Edge effects are eliminated via a window function (often a Hamming window). The statistics (mean, variance, etc.) of these characteristics are then calculated for each frame's tonal texture features.

- **Mel-Frequency Cesptral Coefficients (MFCCs):** The MFCC was developed to record short-term spectral-based attributes. After obtaining

the logarithm of the amplitude spectrum based on the STFT of each frame, the frequency bins are clustered and smoothed in line with Mel frequency scaling designed to match perception.

- **Spectral Centroid:** A measurement known as the spectral centroid is employed in digital signal processing to describe the gravitational centre of a spectrum. A sound signal's "brightness" or "timbre" can be identified in this way.
- **Spectral Rolloff:** Another metric used in digital signal processing to describe a spectrum's form is spectral rolloff. It is described as the frequency below which a specific proportion of the spectrum's overall energy is focused.
- **Spectral Flux:** In digital signal processing, the term "spectral flux" refers to a measurement that expresses the rate of change in a spectrum over time. It measures how much a signal's spectral difference varies between succeeding frames.
- **Zero Crossings:** In digital signal processing, the concept of zero crossing is used to identify the places at which a signal's amplitude crosses zero. It is, in other words, the moment when the signal's sign flips from positive to negative or vice versa.

A musical signal's temporal movement is characterised by the rhythmic content characteristic, which also includes information on rhythmic regularity, beat, pace, and time signature. Usually derived from beat histograms, the set of functions used to depict rhythmic structure is based on the recognition of the signal's most important periodicities. The beat histogram is created by breaking the music signal down into several octave frequency bands and using the time-domain amplitude envelope of each band.

**DWCHs:** A sound file is an illustration of an oscillating waveform in the time domain. According to the equation  $M(t) = D(A, t)$ , the amplitude of a sound file changes with time. where  $A$  is the amplitude, typically between  $[-1, 1]$ , in this case. Identification of amplitude variation is essential for categorising music since it contains the distinguishing qualities. On the one hand, the histogram technique for distribution estimation works well. However, raw signals in the time domain are not a useful representation, particularly for content-based classification, as the most significant features are hidden in the frequency domain. The frequency spectrum of sound, on the other hand, is often divided into octaves, each of which has unique properties. A frequency band's logarithmic connection between arbitrary frequencies with a pitch ratio of 2 to 1 is known as an octave. Excellent temporal and frequency resolution is provided by the wavelet decomposition approach, which is analogous to the perceptually scaled sound octave division model. In other words, wavelet-based audio signal decomposition yields a collection of subband signals with a range of frequencies that correlate to various properties. This encourages us to extract features using the wavelet histogram method. The wavelet coefficients are dispersed over a range of resolutions and frequency ranges.

**Algorithm(s) Used:**

- **Support Vector Machines:** SVMs have demonstrated strong performance in binary classification problems. Support vector machines primary goal is to identify the hyperplane with the greatest margin of separation between positive and negative data points. To expand SVM for multiclass classification, use one-versus-remains, pairwise comparisons, and multiclass objective functions.
- **K-Nearest Neighbor (KNN):** A nonparametric classifier is KNN. Asymptotically, it has been demonstrated that KNN error may be up to twice as high as Bayesian error. KNNs have been used to solve a number of music analysis issues. The core concept is to let a select few

neighbours have an impact on point decisions.

- **Gaussian Mixture Models (GMM):** In order to retrieve musical information, GMM, which models acoustic dispersion, is frequently utilised. They predicated the existence of a probability density function for each class, which may be represented as a combination of many multidimensional Gaussian distributions. Afterward, estimated the parameters for each Gaussian component and mixture weight using an iterative EM technique.
- **Linear Discriminant Analysis (LDA):** In the statistical pattern recognition literature, discriminant analysis approaches are well known for learning discriminant feature transformations. This approach has been used to solve several classification issues. The core objective of LDA is to identify the optimum linear transformation to divide classes, and classification is then performed in the transformed space using metrics like Euclidean distance.

The author of this study suggests DWCH, a novel feature extraction technique, for the categorization of musical genres. The representation of music signals by DWCH substantially enhances classification accuracy by constructing histograms of Daubechies wavelet coefficients in various frequency bands. Additionally, they included a comparison of several feature extraction and classification techniques and looked at how well various techniques performed on various feature sets.

Fourth paper which we reviewed was “**Music Genre Classification With Taxonomy**” by Tao Li and Mitsunori Ogiwara. In their study, they used the concept of taxonomy. There were several reasons at that time why taxonomies are so useful for classifying music genres. Rather than creating generic searches, many users prefer to browse hierarchical catalogues and build queries that match

specific kinds. Utilising taxonomies enhances usability, search success, and user pleasure, according to experiments. Second, taxonomic structures identify dependencies between genera and provide a valuable resource for many questions. Hierarchical structures generally improve the efficiency of both learning and representation. Hierarchical structures enable the use of a divide-and-conquer approach, improving efficiency and accuracy.

Third, taxonomies are more tolerant of classification errors than flat taxonomies. A divide-and-conquer approach concentrates errors at specific levels of the hierarchy.

The author's used two datasets for experiments. The first data set, data set A, contains 1000 songs across 10 genres, with 100 songs for each genre. The 10 genres are blues, classical, country, disco, hip hop, jazz, metal, pop, reggae and rock. 756 sounds from five different genres make up the second dataset, dataset B. These genres include ambient, classical, fusion, jazz, and rock.

The below figures show the taxonomy structure for dataset A and dataset B. One approach to using taxonomies is a top-down level-based approach that arranges clusters in a two-level tree hierarchy and trains a classifier at each inner node.

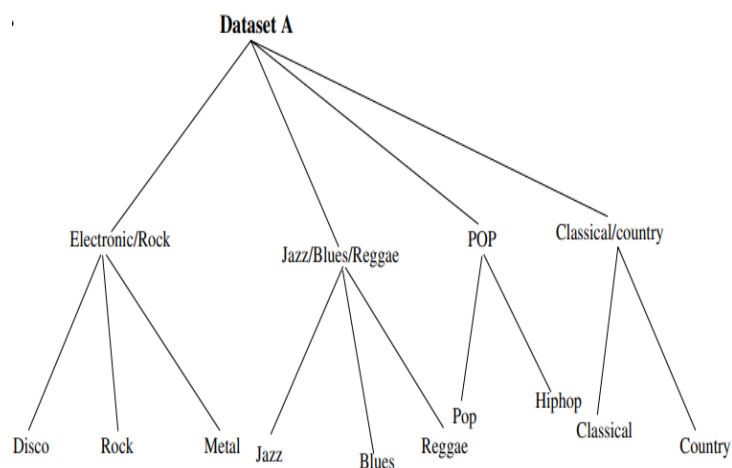


Figure 7: Taxonomy Structure for Dataset A

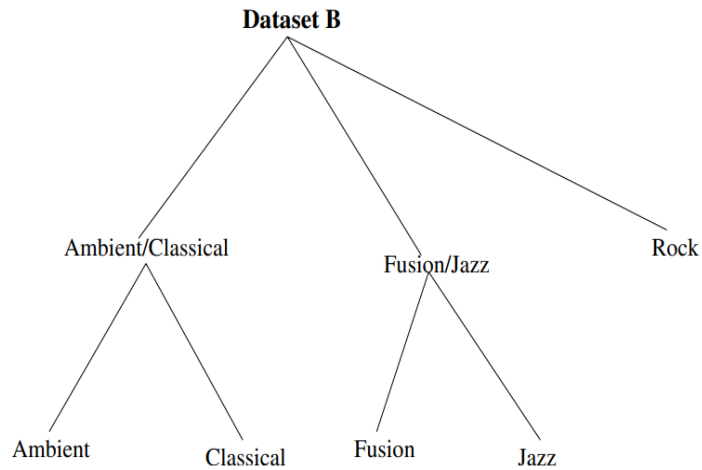


Figure 8: Taxonomy Structure for Dataset B

The algorithm they used to classify the songs was Support Vector Machines. They used linear kernels in their experiments. They used 70% of the data to train the model and rest of the data to test it.

They employed confusion matrices produced by certain effective classifiers to determine genre associations, which was their main method for creating taxonomies. In actuality, the retained validation set was subjected to a classifier to create a confusion matrix. Use a projection with linear discrimination. However, they chose linear discriminant projection because to its effectiveness and accuracy above other classification techniques. SVMs on the other hand, while accurate, have lengthy training durations for multiclass issues.

Coming to the final conclusion of this paper, it explores the use of hierarchical taxonomies in classifying musical genres. In particular, they discussed the rationale for including taxonomies, experimentally assess the implications of using taxonomies, and propose approaches to generate hierarchical taxonomies.

## **Chapter 3: SYSTEM DESIGN & DEVELOPMENT**

The overall aim of this study is to design and develop an algorithm that can successfully perform the task with a high level of accuracy, even when provided with different input files such as audio or song files.

To achieve this goal, will take a step-by-step approach that involves analysing the available datasets, selecting the most appropriate machine learning or deep learning algorithms based on the data analysis, and designing and developing the algorithm using state-of-the-art techniques. The developed algorithm will then be rigorously tested and validated to ensure that it meets our performance criteria and accurately classifies songs into their respective genres.

The successful implementation of this project will have significant implications for the music industry, as it will allow for more accurate and efficient categorization of songs based on their genres. This will, in turn, enhance the user experience for music enthusiasts, improve recommendation systems for music streaming services, and aid in better music marketing and distribution strategies.

### **3.1 Algorithm Analysis**

We have previously seen a number of research papers where the authors provided a variety of ways for tackling this problem, and as a result, we need to develop a classification model that is capable of categorising a given set of data into the numerous categories that are now accessible. Therefore, we may also use a variety of methods based on deep learning and machine learning algorithms. Support Vector Machine (SVM) and Convolutional Neural Network (CNN) are a couple of the Classification techniques that may be used to solve this issue.

### **3.2 Designing of Algorithm**

After all the research work done by us, we chose two classification models to be



implemented and can note their performance or accuracy based on various outputs given by the model when different inputs are given to them. So, the two algorithms are Support Vector Machine (SVM) and Convolutional Neural Network (CNN).

1. Support Vector Machine (SVM):

As we all know that support vector machine or SVM is mostly used for classification of two different classes. But in our project dataset that we have taken (i.e. GTZAN) have 10 different classes, i.e. classical, hip-hop, blues, country, jazz, metal, disco, pop, rock and reggae. But we can use SVM by using the one vs the rest method to achieve our goal.

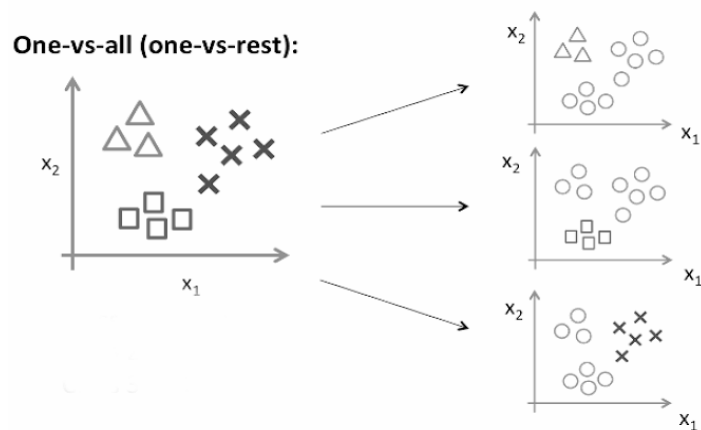


Figure 9: One vs Rest method of SVM

2. Convolutional Neural Network (CNN):

Now we choose convolutional neural network (CNN) as our second model, so for designing the deep learning model we need to firstly extract features from the given audio/song file as input to model with the help of some convolutional networks that are present at various layers of the network. We have to apply max pooling for the purpose of using this is to downsample according to the dimensions of a given input. This activation therefore reduces the number of parameters. This further picks

up the most prominent feature of the previous feature map i.e. calculate the maximum or largest value in each patch of each feature map.

Our input convolves through the layers and then the output layer generated acts as the input layer for the next one, this process goes on. The diagram below represents the flow diagram of the discussed CNN model for genre based classification of songs.

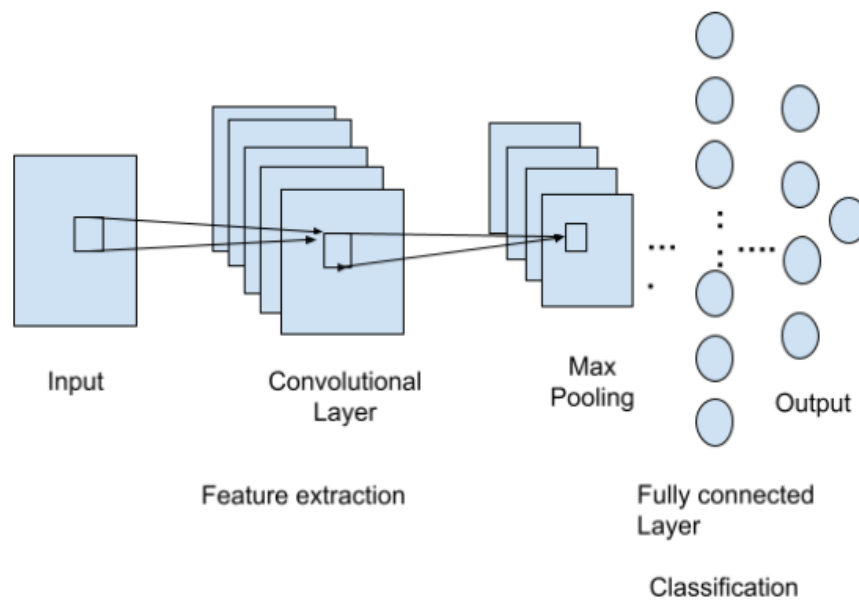


Figure 10: CNN Model

### 3.3 Model Development

Our first step is to understand the dataset, i.e. the GTZAN dataset for further processing. In our project, we have used the GTZAN dataset from kaggle which contains around 1000 samples of songs of 10 different genres. The 10 different genres are: hip-hop, classical, blues, country, jazz, metal, disco, pop, rock and reggae.

The dataset has the following folders:

- Genres original:  
A collection of 10 genres, each with 100 audio files, each 30 seconds long (famous GTZAN dataset, MNIST for sound)
- Images original:  
A depiction of each audio file in graphic form. A neural network may be used to categorise data since they often make use of some sort of picture representation.
- 2 CSV files:  
It includes the attributes of an audio file. The mean and variance calculated from various parameters that may be derived from the audio recording for each song (30 seconds long) are stored in one file. The format of the other file is the same, except the song is divided into 3-second audio files.

Name	Date modified	Type	Size
genres_original	9/28/2022 10:18 PM	File folder	
images_original	9/28/2022 10:18 PM	File folder	
features_3_sec	3/24/2020 2:07 PM	Microsoft Excel Co...	10,816 KB
features_30_sec	3/24/2020 2:07 PM	Microsoft Excel Co...	1,083 KB

Figure 11: Files inside dataset

Now we begin our implementation with importing all necessary libraries like pandas, numpy, matplotlib and so on. Then we import the dataset into our google colab notebook and then with the help of pandas library we import the csv file (features\_3\_sec.csv) into our colab notebook. Then we remove some unwanted columns or features from our dataset like the filename column and clean the data.

As of now we have researched and designed two different models that we are going to implement. These models are: support vector machine (SVM) and convolutional neural networks (CNN).

1. Support Vector Machine (SVM):

In the support vector machine (SVM) model, we first employ kernel functions to convert the input data into processing data, or, more generally, we modify the training data set to allow nonlinear decision surfaces to be converted into linear equations in a higher dimensional space. The following list includes the many kinds of kernel functions that may be used:

a. Gaussian Kernel:

Used to perform transformations without prior knowledge of the data.

Equation for this kernel implementation is as follows:

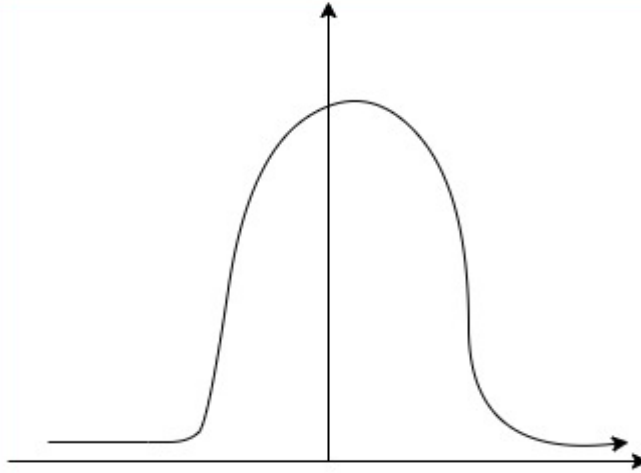
$$K(x, y) = e^{-\left(\frac{\|x-y\|^2}{2\sigma^2}\right)} \text{-----Eq. 3.1}$$

b. Gaussian Kernel Radial Basis Function (RBF):

It is the same as the kernel function above, but uses the radial base method for better conversion.

Equation is as follows:

$$\begin{aligned} K(x, y) &= e^{-\gamma\|x - y\|^2} \\ K(x, x1) + K(x, x2) & \text{(Simplified - Formula)} \\ K(x, x1) + K(x, x2) & > 0 \text{(Green)} \\ K(x, x1) + K(x, x2) & = 0 \text{(Red)} \end{aligned} \text{-----Eq. 3.2}$$



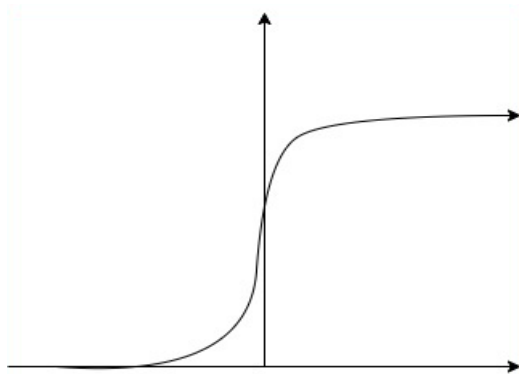
Graph 1: Gaussian Kernel Graph

c. Sigmoid Kernel:

This function is equivalent to the dual-level perceptron neural network model used as the activation function of artificial neurons of the network.

Equation of sigmoid kernel :

$$K(x, y) = \tanh(\gamma \cdot x^T y + r) \text{ -----Eq. 3.3}$$



Graph 2: Sigmoid Kernel Function

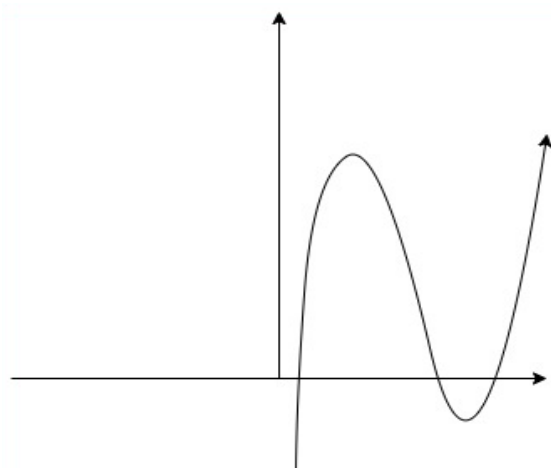
d. Polynomial Kernel:

It represents the similarity of the vectors in the training data set in the feature space to the polynomials of the original variables used in the kernel.

Equation for polynomial function is as follows:

$$K(x, y) = \tanh(\gamma \cdot x^T y + r)^d, \gamma > 0$$

-----Eq. 3.4



Graph 3: Polynomial Kernel Function

In our project we implemented the support vector machine (SVM) model using the polynomial kernel and the model came up with an accuracy of 89.35% which is very less compared to the CNN model.

2. Convolutional Neural Network (CNN):

Now we begin with the implementation of the CNN model for our genre based classification of songs. After finding the perfect dataset and importing it into our project, and before building the model we firstly need to learn how we can handle and visualize audio files (.wav extension) for our genre based classification model. And for this purpose we have used several libraries and they are as follows:

- a. Librosa: A Python tool or library for analysing audio or music/songs is called Librosa.
- b. `Python.display.audio`: It let us play audio directly in an IPython notebook.

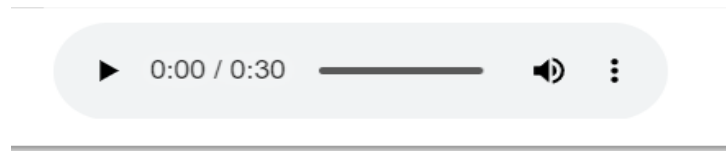


Figure 12: audio file available on colab

Some methods that we can use for the visualization purpose of audio or song files on colab notebooks are as follows:

- a. By plotting Raw Wave files:

With time plotted on the x-axis and amplitude plotted on the y-axis, a waveform is a graphic representation of sound. It helps us quickly analyse the audio data and identify which genres may be more similar than others through visual comparison and contrast.

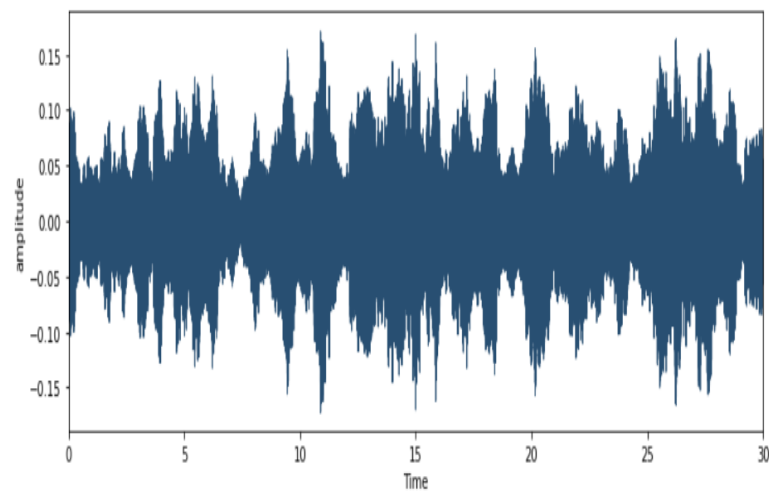


Figure 13: Raw Wave plot of audio file

b. Spectral Rolloff:

It is the frequency at which a given percentage of the total spectral energy of the wave of the audio file (for eg 85% is lying).

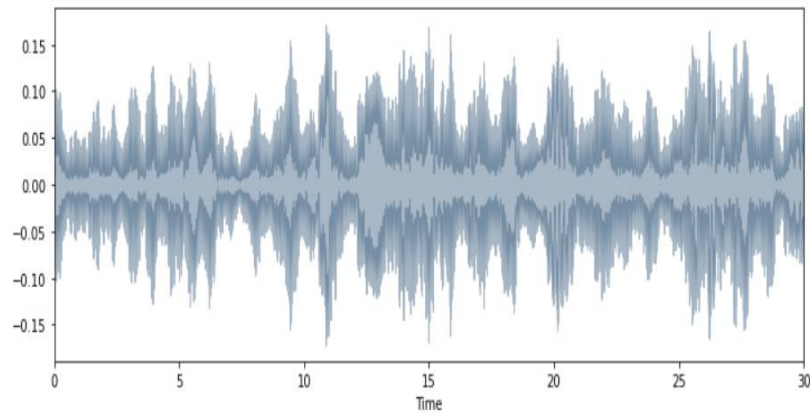


Figure 14: Spectral Rolloff of audio file

c. Spectrograms:

A spectrogram is a visual representation of a signal's strength over time at different frequencies included in a certain waveform. We can see variations in the energy level over time in addition to more or less energy, such as 2 Hz vs. 10 Hz.

The spectrogram is sometimes referred to as voicegrams, sonographs, or voiceprints.

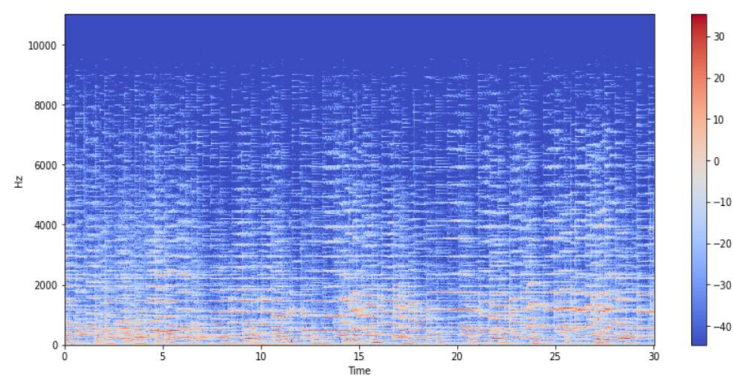


Figure 15: Spectrogram of audio file



d. Chroma Features:

It is a powerful tool for analysing musical characteristics that can meaningfully classify pitches and whose tuning is close to temperamental scale.

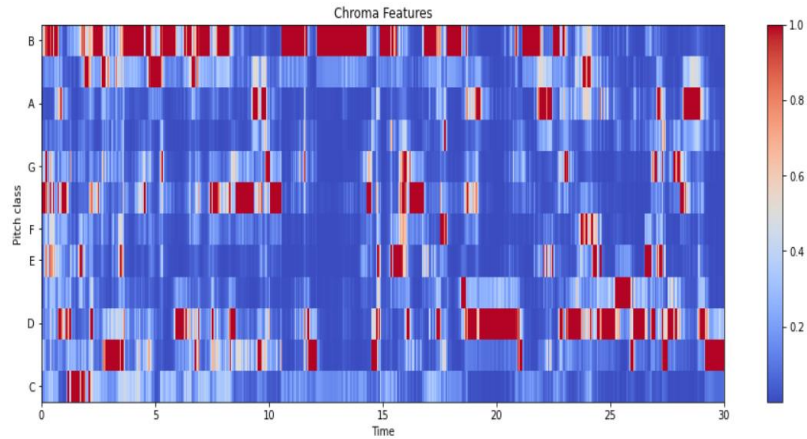


Figure 16: Chroma Feature of audio file

e. Zero Crossing Rate:

Zero crossings are said to be occurring when successive samples have different algebraic signs. The frequency component of a signal can be easily measured using the zero-crossing ratio.

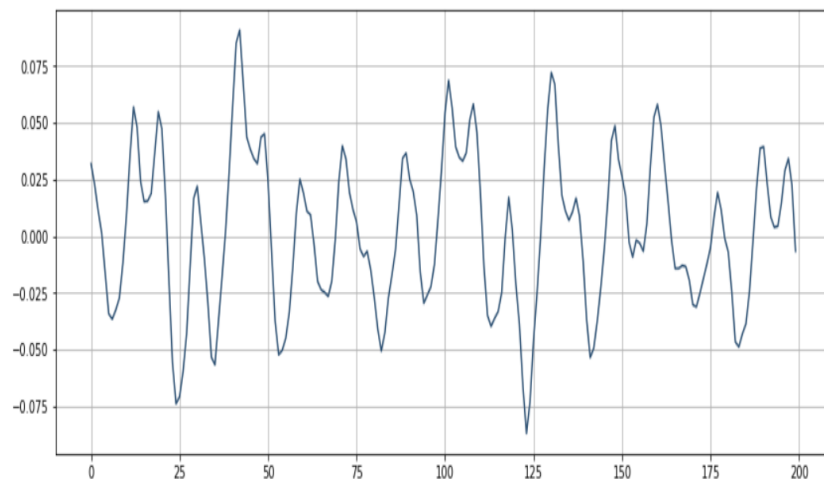


Figure 17: Zero Crossing Rate of audio file

Prior to final data training, data preparation is necessary. Use the LabelEncoder() method of sklearn. pre-processing to encode the last column, "label."

Text cannot be present in the data if we are trying to run a model. As a result, prior to running our model, we must prepare this data. The Label Encoder class may be used to translate this type of category text data into numeric information that the model can read.

After this we use a standard scaler to standardize features by removing the mean and scaling by unit variance. The standard sample score  $x$  is calculated as:

$$z = (x - \mu)/s \text{ -----Eq. 3.5}$$

Standardisation of data sets is frequently necessary for machine learning estimators. If the data doesn't appear to be regularly distributed, it could not be performing as expected by our model for categorising the genres of music.

After this we divide our dataset into training and testing set in the ratio of 8:2.

And now we start with the final part of our project that is the genre classification of different songs. After features were taken out of the raw data, the model needed to be trained. Training a model may be done in a variety of ways. Several of these strategies consist of:

- a. Multiclass Support Vector Machine (SVM)
- b. Convolutional Neural Network (CNN)

We'll now train a model in this project using the CNN technique. We picked this strategy because many types of study have indicated that it produces the greatest outcomes for this issue.

We utilised the Adam optimizer to train the CNN model. The training model's selected epoch is 600. After comparing many optimizers, we settled on the Adam optimizer since it produced the best results.

An algorithm used in the gradient descent optimisation method is adaptive moment estimation. When dealing with complex situations involving vast quantities of data or parameters, this approach is particularly effective.

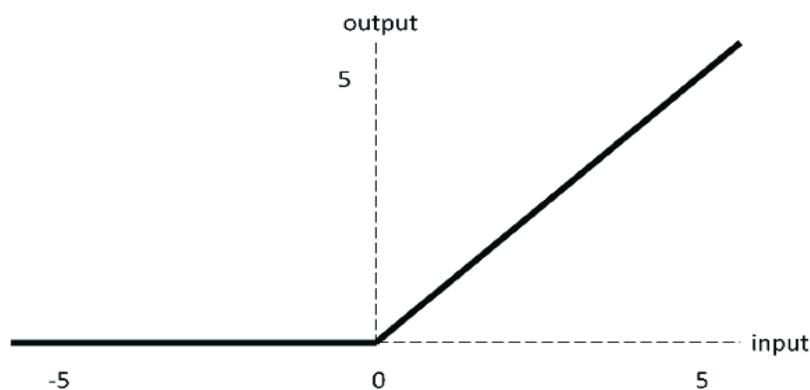
There are many activation functions present like:

- a. Linear Activation Function
- b. Non-linear Activation Functions
- c. Sigmoid
- d. Tanh
- e. RELU
- f. Leaky RELU

But in our model all the different hidden layers use the RELU activation function and the output layer uses the softmax function.

Equation for RELU is as follows:

$$f(x) = \max(0, x) \text{ -----Eq. 3.6}$$



Graph 4: RELU Activation Function

The `sparse_categorical_crossentropy` function is used to calculate loss. Additionally, dropouts are employed to stop overfitting.

More epochs can increase the accuracy of the model, but the threshold may be reached after a certain amount of time, so the value should be determined accordingly.

Thus we complete our model and get a good accuracy from our model of 93.29%.

The provided flow chart encapsulates the entire process of constructing a machine learning model, commencing from the importation of the GTZAN dataset. The model is designed to minimise overfitting using dropout techniques, while the loss function is computed using sparse categorical cross-entropy. The presented flow chart summarises the sequential steps of this process in detail.

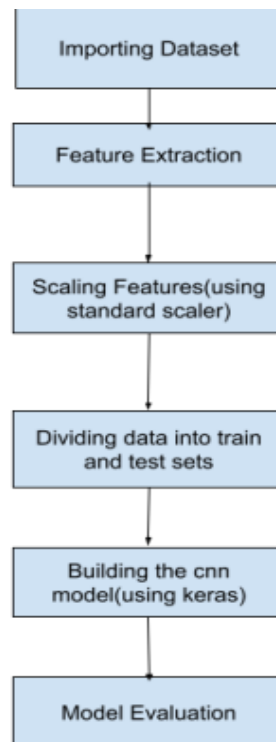


Figure 18: Final Flow Diagram of our CNN Model

## Chapter 4: EXPERIMENTS & RESULTS ANALYSIS

### 4.1 Requirements

#### 4.1.1 Language Used

Python is the language being utilised in this project. Deep learning and machine learning are the fields in which this phrase is employed. Machine learning (ML), which is a subset of artificial intelligence (AI), includes deep learning. Making robots and computers think and reason similarly to human brains is the aim. In essence, it imitates how a person might learn new knowledge.

Python can create a wide variety of different data visualisations, including line charts, bar charts, pie charts, histograms, and 3D charts. Python also has numerous libraries, such as TensorFlow and Keras, that enable programmers to write data analysis and machine learning programs faster and more efficiently.

Python offers packages and modules that encourage the modularity and reuse of code in programmes. On all major systems, the Python interpreter and comprehensive standard library are freely distributable in source or binary form.

#### 4.1.2 Libraries Used

- pandas:  
It is a Python library that offers a quick, adaptable, and expressive data structure to make manipulating "relational" or "labelled" data simple and natural. It seeks to serve as a basic, high-level building block for Python's use in actual, real-world data analysis.
- numpy:  
It is a general-purpose software for array processing. high-performance

multidimensional array objects and tools for working with such arrays are made available. Python's fundamental scientific computing module. The programme is open source.

- matplotlib:

It is a comprehensive library for creating static, animated and interactive visualisations in Python. Matplotlib makes simple things easy and hard things possible.

- scipy:

It is a library for scientific computing that is built on NumPy. Scientific Python is a common abbreviation. more beneficial features for signal processing, analytics, and optimisation are provided.

- pickle:

Pickle in Python is primarily used to serialize and deserialize Python object structures. That is, the process of converting a Python object into a byte stream to store in a file/database, maintain program state between sessions, transfer data over the network, etc.

- librosa:

Librosa is a Python package for analysing music and audio files. Librosa is primarily used when working with audio data, such as music generation (using LSTM) and automatic speech recognition. It provides the building blocks necessary to create a system for retrieving music information.

- scikit-learn:

scikit-learn is an open-source Python library that implements a set of machine learning, preprocessing, cross-validation, and visualisation algorithms through a unified interface.

- tensorflow:  
TensorFlow is an open-source, Python-friendly numerical library that makes machine learning and neural network development faster and easier.

### 4.1.3 System Requirements

Tools and technologies to be used:

1. Machine Learning and Deep Learning

Specific software Requirement:

1. Jupyter Notebook/Google Colaboratory.

### 4.1.4 Hardware Requirements

- Ram: 8GB or higher,
- Storage: 500GB,
- CPU: 2GHz or faster, and
- Architecture: 32Bit or 64Bit.

## 4.2 Results at various stages

Now let us see snapshots/outputs of our project or genre base song classification at various stages. Firstly let us analyse the dataset:

1. Since our first step was to read the dataset and import it to our colab notebook using pandas library and make a dataframe out of it so that we can do further calculations and preprocessing over the given dataset.

	filename	length	chroma_stft_mean	chroma_stft_var	rms_mean	rms_var	spectral_centroid_mean	spectral_centroid_var	spectral_bandwidth_mean	spect
0	blues.00000.0.wav	66149	0.335406	0.091048	0.130405	0.003521	1773.065032	167541.630869	1972.744388	
1	blues.00000.1.wav	66149	0.343065	0.086147	0.112699	0.001450	1816.693777	90525.690866	2010.051501	
2	blues.00000.2.wav	66149	0.346815	0.092243	0.132003	0.004620	1788.539719	111407.437613	2084.565132	
3	blues.00000.3.wav	66149	0.363639	0.086856	0.132565	0.002448	1655.289045	111952.284517	1960.039988	
4	blues.00000.4.wav	66149	0.335579	0.088129	0.143289	0.001701	1630.656199	79667.267654	1948.503884	

5 rows x 60 columns

Figure 19: First 5 rows of dataset

2. Now we should analyse the dataset.

```
filename_          object
length            int64
chroma_stft_mean  float64
chroma_stft_var   float64
rms_mean          float64
rms_var           float64
spectral_centroid_mean float64
spectral_centroid_var float64
spectral_bandwidth_mean float64
spectral_bandwidth_var float64
rolloff_mean      float64
rolloff_var       float64
zero_crossing_rate_mean float64
zero_crossing_rate_var float64
harmony_mean      float64
harmony_var       float64
perceptr_mean     float64
perceptr_var      float64
tempo             float64
mfcc1_mean        float64
mfcc1_var         float64
mfcc2_mean        float64
mfcc2_var         float64
mfcc3_mean        float64
mfcc3_var         float64
mfcc4_mean        float64
mfcc4_var         float64
mfcc5_mean        float64
mfcc5_var         float64
mfcc6_mean        float64
mfcc6_var         float64
mfcc7_mean        float64
mfcc7_var         float64
mfcc8_mean        float64
mfcc8_var         float64
mfcc9_mean        float64
mfcc9_var         float64
mfcc10_mean       float64
mfcc10_var        float64
mfcc11_mean       float64
mfcc11_var        float64
mfcc12_mean       float64
mfcc12_var        float64
mfcc13_mean       float64
```

Figure 20: Columns available in the dataset

3. Since, the first column “filename” is of no use to us, so we dropped it.
4. Lets see the different labels or the genres that are present in the dataset.

```
['blues' 'classical' 'country' 'disco' 'hiphop' 'jazz' 'metal' 'pop'
 'reggae' 'rock']
```

Figure 21: Labels/Genres in Dataset

5. Now let us see the importing of an audio file and visualising it at various different stages. We import an audio file in colab notebook using the “librosa” library present in python. For example, Firstly we import a “Pop” song in our colab. The below visualisations are for the uploaded song:



```
<class 'numpy.ndarray'> <class 'int'>
```

Figure 22: Data Type of audio after loading

```
(array([0.6824313 , 0.545749 , 0.32345143, ..., 0.21088526, 0.14528655,  
0.          ], dtype=float32), 45600)
```

Figure 23: Features extracted after loading in colab using Librosa

6. Now let us start visualising the song file that we have imported , i.e, the pop genre song. We will visualise the audio using different methods and they are listed below with their particular outputs:

- a. Wave plot:

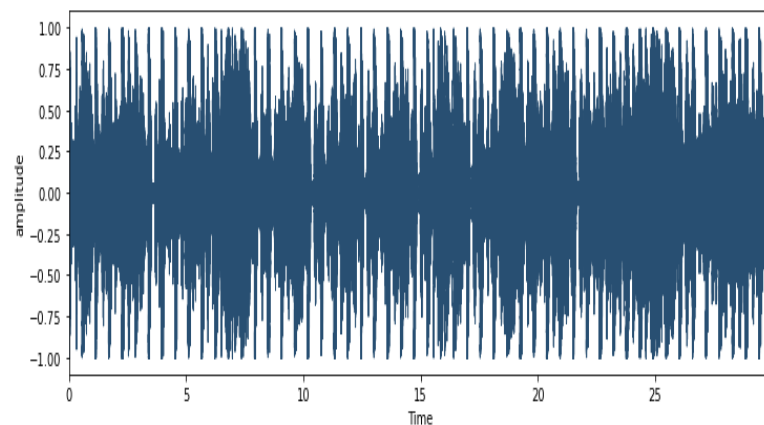


Figure 24: Wave plot for Pop song

This is a raw plot that is plotted for a visual comparison among different songs of different genres. By this plot it is easy to visually differentiate the songs based on amplitude of songs.

b. Spectrogram:

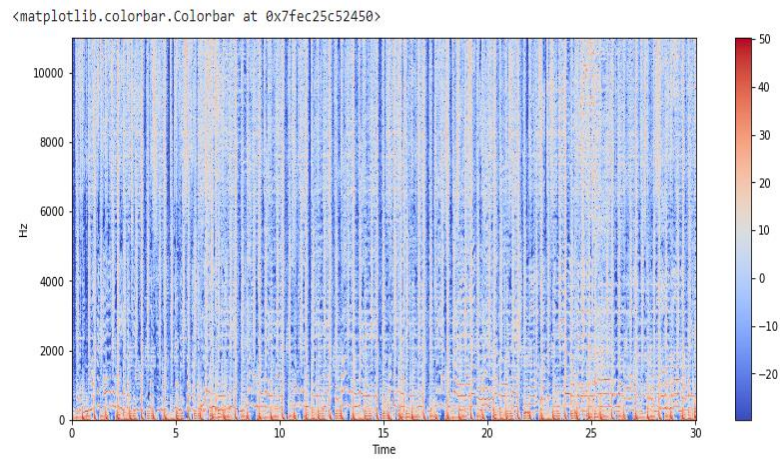


Figure 25: Spectrogram for Pop song

This spectrogram for the uploaded pop song tells the visual representation of the loudness present in the audio at a frequency range from 0 to 10kHz.

c. Spectral Rolloff:

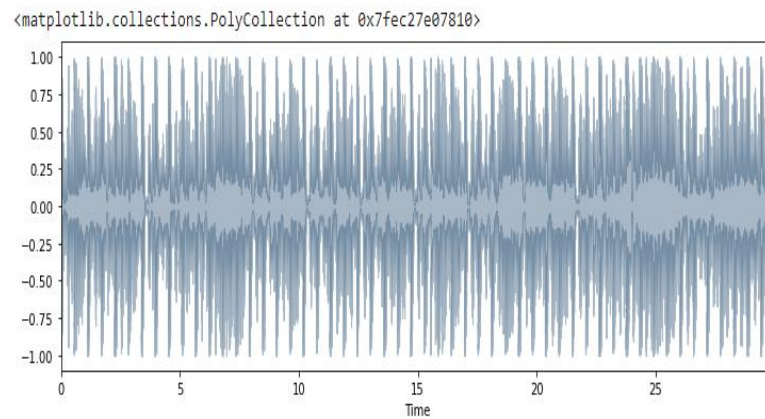


Figure 26: Spectral Rolloff of uploaded Pop song

d. Chroma Features:

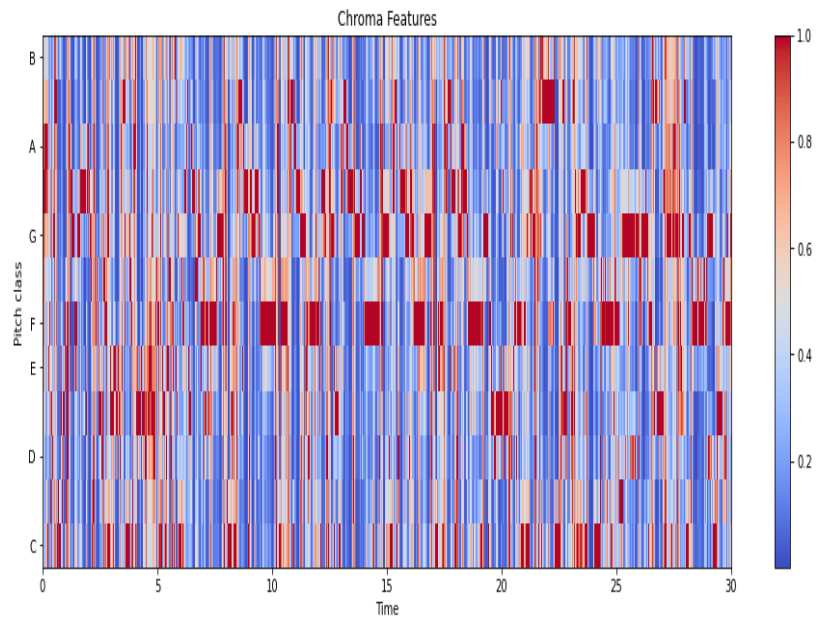


Figure 27: Chroma Feature plot of Pop Song

e. Zero Crossing Rate:

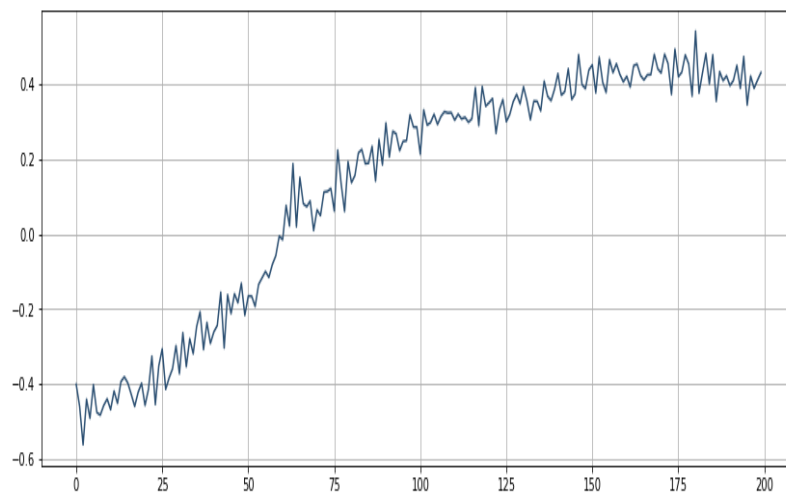


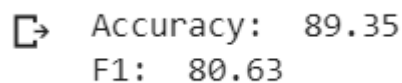
Figure 28: Zero Crossing Rate Plot of Pop Song

Here, in the plot it is very clearly visible that the zero crossing rate of Pop song is “2”.

Now , since we have seen in the study about all the implementation and design part. So, now let us see how well the model has performed on our inputs of different songs and audio files. First of all let us see the performance analysis of SVM.

1. Support vector machine (SVM):

Here, after importing all the libraries and building the SVM model for our genre based classification system, we do prediction on the model. The results/accuracy of the prediction by the model is shown below in the image.



```
↳ Accuracy: 89.35
   F1: 80.63
```

Figure 29: SVM model scores

In the above image we can see that the accuracy of our SVM model is 89.35%. Its F1 score is 80.63 (Primarily used to review accuracy given by classifiers). We can also see the confusion matrix for the model.

2. Convolutional Neural Network (CNN):

- a. Firstly, we imported “Sequential” from keras.
- b. Dataset was split into training and testing with 80% data for training and the remaining data for testing.
- c. Then, we defined a function trainModel which takes three inputs i.e. model, optimizer, epochs. The batch size was set to 128 and the total number of epochs was 600.

```

Epoch 1/600
63/63 [=====] - 2s 14ms/step - loss: 1.5784 - accuracy: 0.4247 - val_loss: 1.0726 - val_accuracy: 0.6271
Epoch 2/600
63/63 [=====] - 1s 11ms/step - loss: 1.0838 - accuracy: 0.6187 - val_loss: 0.8578 - val_accuracy: 0.7082
Epoch 3/600
63/63 [=====] - 1s 11ms/step - loss: 0.8817 - accuracy: 0.7031 - val_loss: 0.7328 - val_accuracy: 0.7678
Epoch 4/600
63/63 [=====] - 1s 12ms/step - loss: 0.7563 - accuracy: 0.7409 - val_loss: 0.6677 - val_accuracy: 0.7833
Epoch 5/600
63/63 [=====] - 1s 11ms/step - loss: 0.6651 - accuracy: 0.7780 - val_loss: 0.6361 - val_accuracy: 0.7848
Epoch 6/600
63/63 [=====] - 1s 10ms/step - loss: 0.5860 - accuracy: 0.8073 - val_loss: 0.5603 - val_accuracy: 0.8118
Epoch 7/600
63/63 [=====] - 1s 11ms/step - loss: 0.5161 - accuracy: 0.8257 - val_loss: 0.5276 - val_accuracy: 0.8263
Epoch 8/600
63/63 [=====] - 1s 11ms/step - loss: 0.4731 - accuracy: 0.8435 - val_loss: 0.4924 - val_accuracy: 0.8343
Epoch 9/600
63/63 [=====] - 1s 10ms/step - loss: 0.4241 - accuracy: 0.8602 - val_loss: 0.4704 - val_accuracy: 0.8418
Epoch 10/600
63/63 [=====] - 1s 11ms/step - loss: 0.3900 - accuracy: 0.8697 - val_loss: 0.4342 - val_accuracy: 0.8594
Epoch 11/600
63/63 [=====] - 1s 11ms/step - loss: 0.3549 - accuracy: 0.8849 - val_loss: 0.4119 - val_accuracy: 0.8699
Epoch 12/600
63/63 [=====] - 1s 11ms/step - loss: 0.3160 - accuracy: 0.8928 - val_loss: 0.4149 - val_accuracy: 0.8729
Epoch 13/600
63/63 [=====] - 1s 11ms/step - loss: 0.2982 - accuracy: 0.9024 - val_loss: 0.3945 - val_accuracy: 0.8789
Epoch 14/600
63/63 [=====] - 1s 11ms/step - loss: 0.2726 - accuracy: 0.9040 - val_loss: 0.3774 - val_accuracy: 0.8824

```

Figure 30: Running 600 Epochs

- d. Then, finally the data was trained and the below output shows the training of our data.

```

Model: "sequential"

```

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 512)	30208
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131328
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 128)	32896
dropout_2 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 64)	8256
dropout_3 (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 10)	650

```

=====
Total params: 203,338
Trainable params: 203,338
Non-trainable params: 0
None

```

Figure 31: Epoch training of model

- e. After successfully training the model, we are now ready to evaluate our model for different inputs.

```
Accuracy: 93.29329133033752
Test Loss 0.4837329685688019
```

Figure 32: Accuracy of CNN Model

- f. For evaluation, we give a random index of a song file from the dataset (testing set) to our model and check whether the prediction is correct or not.

```
1/1 [=====] - 0s 105ms/step
Expected: classical,
Predicted label: ['classical']
```

Figure 33: Accurate prediction on sample data

Here , we can see that our model is working fine and it has predicted the correct genre of the randomly selected index, i.e. “classical”.

- g. Then for testing our model, we gave a random .wav file as input and checked whether the model is predicting correct output or not. Here, for example we have uploaded a song of “reggae” genre to see if our model can predict the genre.

```
Choose Files reggae.00001.wav
• reggae.00001.wav(audio/wav) - 1323632 bytes, last modified: 3/24/2020 - 100% done
Saving reggae.00001.wav to reggae.00001.wav
```

Figure 34: A reggae genre song file uploaded

- h. After uploading the data, we extract all the features from the audio which are required by the model for prediction. This is

done with the help of librosa library. After this we predict the output by the help of these features as input to the model.

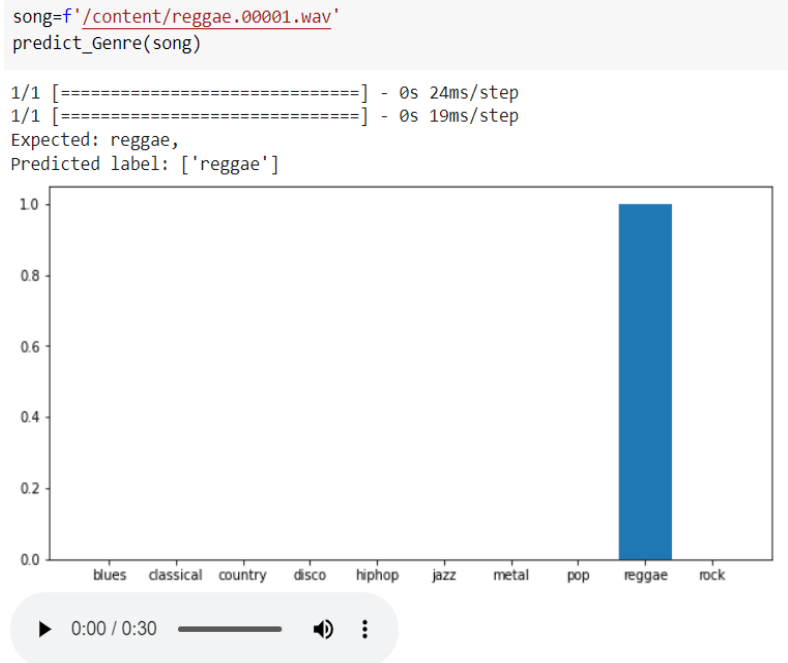


Figure 35: Reggae genre song correctly identified as “Reggae”

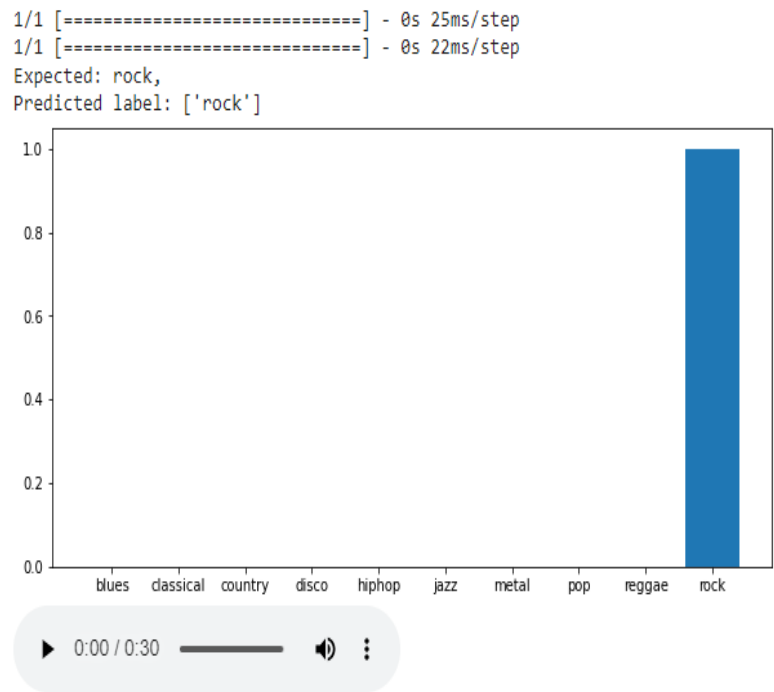


Figure 36: Another Rock genre song correctly identified as “Rock”

Table 3: Accuracy Results of Models Implemented

<b>Model</b>	<b>Accuracy</b>
Support Vector Machine (SVM)	89.35%
Convolutional Neural Network (CNN)	93.29%

So, from the above table we can clearly see that CNN gives us better results over the SVM model.



## Chapter 5: CONCLUSIONS

### 5.1 Conclusions

The evaluation metric used in our project is accuracy, which measures the percentage of predicted outputs that match the actual outputs. In our case, the predicted output refers to the genre of the song.

The project's objective was to classify songs or music into several genres using both traditional machine learning models and deep learning. A review of relevant literature was used to demonstrate the efficacy of these classifiers and to establish a benchmark for comparison with the findings of this investigation. We employed two methods: support vector machines and convolutional neural networks.

In our proposed work, we used the GTZAN dataset and built several models to accomplish the task of the genre classification for songs.

Firstly, we choose to employ support vector machines since they are frequently favoured because they may produce accurate results with little processing effort. SVM, or Support Vector Machine, is a tool that may be used for both classification and regression problems.

After implementing the SVM model effectively, we switched to the deep learning model that had been researched in earlier studies. The deep learning models produced superior outcomes and more accurately categorised music. Convolutional neural networks (CNN) were employed, and Adam Optimizer was used to train the model. We used several optimizers, but Adam Optimizer produced the best outcomes.

We used 80% of the available data to train our model and 20% of the data to test our data. Previously we were using 70% to train the data and the remaining 30%

to test the data, but were getting less accuracy.

A method for transforming data into the format required for data processing is known as a kernel function. The kernel function, in general, transforms the training data set to turn nonlinear decision surfaces into linear equations in a higher dimensional space.

The accuracy for the SVM model comes out to be 89.35% this gives us a conclusion that SVM can be used in case of classification.

Clearly, we can see through the results of SVM that the accuracy of the model is not that high, as it has more noisy data and target classes are overlapping each other.

Now, coming to the results of the convolutional neural network (CNN) model, it has an accuracy of 93.29%. Therefore, we conclude that using the concept of neural networks we get higher accuracy for developing the model and then testing it.

The only limitation of our project is that it cannot work efficiently if the timestamp of audio files are increased, and our models (both SVM and convolutional neural network) are applied to classify music. The reason can be due to the mix of moods represented by the song and all the different categories of features appearing. Also in the case of large lyrical music audio files, the model fails to precisely predict the category of music.

## **5.2 Future Work**

In our proposed work, we are able to achieve an accuracy of approximately 93% in classifying music genres using machine learning algorithms. However, they have suggested that further improvements could be made by applying various different techniques and algorithms to the system.

The same methods used in this study may be used to categorise music based on additional labels, such as artist. Furthermore, it could be feasible to develop a system for classifying music moods by adding other metadata text aspects like album, song title, or lyrics.

The automatic production of a selection of appropriate images for every particular song is one potential use for the music-image mapping technique. This might take the role of manual compilations in YouTube videos or abstract colour animations in media players, giving viewers a more individualised and interesting experience.

As new types of music and audio formats emerge in the future, it will be necessary to collect new data and create datasets to apply classification models accordingly. To build upon the results of this work, a logical next step would be to develop a hierarchical genre classification model that can use higher-level meta genre determination to improve accuracy at lower levels.

Future work could also explore the relationship between genre and mood, and investigate how the two can be integrated into a unified classification system. An important consideration for future research is the use of annotated metadata, including genres, moods, grooves, composers, performers, lyrics, time signatures, chord progressions, and instruments present. Collaboration efforts to construct high-quality ground truth data would be necessary to ensure the reliability of the classification models.

In the future, deep learning approaches could also be applied to improve the accuracy of the classification models. By optimising the parameters of the deep learning models, it may be possible to achieve higher accuracy and more precise predictions. Ultimately, the development of advanced classification models will enhance the user experience and enable personalised music recommendations.

based on individual preferences and mood.

### **5.3 Application of the Project**

Music genre classification has become an essential tool for various music applications, including Drinkify and Pandora, as well as music streaming services like Spotify, Gana, and Apple Music. These platforms use machine learning algorithms to classify music based on various features, such as rhythm, melody, and timbre, among others. By analysing these features, music genre classification can predict the most appropriate music for different purposes, including relaxation, focus, and entertainment.

Moreover, music genre classification can also be used in music therapy, where the appropriate music can be used to treat certain conditions, such as anxiety, depression, and pain management. By classifying music based on its emotional and psychological impact, music therapists can choose the most effective music for their patients.

The application of music genre classification is not limited to entertainment or therapy; it can also be used in various industries, such as advertising and marketing. By understanding the music preferences of their target audience, advertisers and marketers can create targeted campaigns that resonate with their customers.

In conclusion, music genre classification has become an essential tool in various music applications and industries. With the help of machine learning algorithms, music can be classified based on its various features, and this information can be used to enhance user experience, treat certain conditions, and create targeted marketing campaigns.

## REFERENCES

- [1] Li, T., Ogihara, M., & Li, Q. (2003). A comparative study on content-based music genre classification. *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '03*, 282.
- [2] Tao Li, & Ogihara, M. (n.d.). Music Genre Classification with Taxonomy. *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, 197–200.
- [3] Ndou, N., Ajoodha, R., & Jadhav, A. (2021). Music Genre Classification: A Review of Deep-Learning and Traditional Machine-Learning Approaches. *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, 1–6.
- [4] Ghildiyal, A., Singh, K., & Sharma, S. (2020). Music Genre Classification using Machine Learning. *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 1368–1372.
- [5] Shah, M., Pujara, N., Mangaroliya, K., Gohil, L., Vyas, T., & Degadwala, S. (2022). Music Genre Classification using Deep Learning. *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*, 974–978.
- [6] Nam, J., Choi, K., Lee, J., Chou, S.-Y., & Yang, Y.-H. (2019). Deep Learning for Audio-Based Music Classification and Tagging: Teaching Computers to Distinguish Rock from Bach. *IEEE Signal Processing Magazine*, 36(1), 41–51.
- [7] Rajanna, A. R., Aryafar, K., Shokoufandeh, A., & Ptucha, R. (2015). Deep Neural Networks: A Case Study for Music Genre Classification. *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*,

655–660.

[8] Senac, C., Pellegrini, T., Mouret, F., & Pinquier, J. (2017). Music Feature Maps with Convolutional Neural Networks for Music Genre Classification. *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*, 1–5.

[9] Rao, M. S., Pavan Kalyan, O., Kumar, N. N., Tasleem Tabassum, Md., & Srihari, B. (2021). Automatic Music Genre Classification Based on Linguistic Frequencies Using Machine Learning. *2021 International Conference on Recent Advances in Mathematics and Informatics (ICRAMI)*, 1–5.

## APPENDICES

1. Convolutional Neural Network (CNN) Model for Genre Classification of songs:

```
#Importing Sequential from keras
from keras.models import Sequential

#Function to train model
def trainModel(model, epochs,optimizer):
    batch_size=128
    model.compile(optimizer=optimizer,
                  loss='sparse_categorical_crossentropy',
                  metrics='accuracy')
    return model.fit(X_train,y_train,validation_data=(X_test, y_test),
                    epochs=epochs,
                    batch_size=batch_size)

def plotValidate(history):
    print("Validation Accuracy", max(history.history["val_accuracy"]))
    pd.DataFrame(history.history).plot(figsize=(12,6))
    plt.show()

#Building of CNN layers
model= keras.models.Sequential([
    keras.layers.Dense(512,activation='relu',
input_shape=(X_train.shape[1],)),
    keras.layers.Dropout(0.2),

    keras.layers.Dense(256,activation='relu'),
    keras.layers.Dropout(0.2),

    keras.layers.Dense(128,activation='relu'),
    keras.layers.Dropout(0.2),
```

```
keras.layers.Dense(64,activation='relu'),
keras.layers.Dropout(0.2),

keras.layers.Dense(10,activation='softmax'),
```

)

```
#Printing model summary
print(model.summary())
#training on 600 epochs
model_history =
trainModel(model=model,epochs=600,optimizer='adam')
```

2. Function to predict a single sample using the model:

```
def predict(model, X, y):
    """
    :param model: Trained classifier
    :param X: Input data
    :param y (int): Target
    """

    # add a dimension to input data for sample - model.predict() expects
    a 4d array in this case
    X = X[np.newaxis, ...] # array shape (1, 130, 13, 1)

    # perform prediction
    prediction = model.predict(X)

    # get index with max value
    predicted_index = np.argmax(prediction, axis=1)

    print("Expected: {}, \nPredicted label: {}".format(lb[y],
lb[predicted_index]))
```



3. Function to extract features from audio and predict the genre:

```
def predict_Genre(song):
    y, s = librosa.load(song)
    trim_y, _ = librosa.effects.trim(y)
    "It will trim leading and trailing silence from an audio signal.
In this code, we will remove audio signal that is lower than 10db"
    chroma_stft = librosa.feature.chroma_stft(y=trim_y,
sr=s,n_fft=2048, hop_length=512).flatten()
    rmse = librosa.feature.rms(y=trim_y, frame_length=2048,
hop_length=512).flatten()
    spec_cent = librosa.feature.spectral_centroid(y=trim_y, sr=s,
n_fft=2048,hop_length=512).flatten()
    spec_bw = librosa.feature.spectral_bandwidth(y=trim_y,
sr=s,n_fft=2048, hop_length=512).flatten()
    rolloff = librosa.feature.spectral_rolloff(y=trim_y + 0.01,
sr=s,n_fft=2048, hop_length=512).flatten()
    zcr = librosa.feature.zero_crossing_rate(trim_y,frame_length=2048,
hop_length=512).flatten()
    y_harmonic, y_percep = librosa.effects.hpss(trim_y)
    tempo,beats = librosa.beat.beat_track(y, sr = s)
    mfcc = librosa.feature.mfcc(y=trim_y, sr=s,win_length=2048,
hop_length=512)
    mfcc_mean=mfcc.T.mean(axis=0)
    mfcc_var=mfcc.T.var(axis=0)

    length=y.shape[0]/1000000000000000
    chroma_stft_mean=chroma_stft.mean()
    chroma_stft_var=chroma_stft.var()
    rms_mean=rmse.mean()
    rms_var=rmse.var()
    spectral_centroid_mean=spec_cent.mean()
    spectral_centroid_var=spec_cent.var()
    spectral_bandwidth_mean=spec_bw.mean()
```

```
spectral_bandwidth_var=spec_bw.var()
rolloff_mean=rolloff.mean()
rolloff_var=rolloff.var()
zero_crossing_rate_mean=zcr.mean()
zero_crossing_rate_var=zcr.var()
harmony_mean=y_harmonic.mean()
harmony_var=y_harmonic.var()
perceptr_mean=y_percep.mean()
perceptr_var=y_percep.var()
mfcc1_mean=mfcc_mean[0]
mfcc1_var=mfcc_var[0]
mfcc2_mean=mfcc_mean[1]
mfcc2_var=mfcc_var[1]
mfcc3_mean=mfcc_mean[2]
mfcc3_var=mfcc_var[2]
mfcc4_mean=mfcc_mean[3]
mfcc4_var=mfcc_var[3]
mfcc5_mean=mfcc_mean[4]
mfcc5_var=mfcc_var[4]
mfcc6_mean=mfcc_mean[5]
mfcc6_var=mfcc_var[5]
mfcc7_mean=mfcc_mean[6]
mfcc7_var=mfcc_var[6]
mfcc8_mean=mfcc_mean[7]
mfcc8_var=mfcc_var[7]
mfcc9_mean=mfcc_mean[8]
mfcc9_var=mfcc_var[8]
mfcc10_mean=mfcc_mean[9]
mfcc10_var=mfcc_var[9]
mfcc11_mean=mfcc_mean[10]
mfcc11_var=mfcc_var[10]
mfcc12_mean=mfcc_mean[11]
mfcc12_var=mfcc_var[11]
mfcc13_mean=mfcc_mean[12]
mfcc13_var=mfcc_var[12]
```

```

mfcc14_mean=mfcc_mean[13]
mfcc14_var=mfcc_var[13]
mfcc15_mean=mfcc_mean[14]
mfcc15_var=mfcc_var[14]
mfcc16_mean=mfcc_mean[15]
mfcc16_var=mfcc_var[15]
mfcc17_mean=mfcc_mean[16]
mfcc17_var=mfcc_var[16]
mfcc18_mean=mfcc_mean[17]
mfcc18_var=mfcc_var[17]
mfcc19_mean=mfcc_mean[18]
mfcc19_var=mfcc_var[18]
mfcc20_mean=mfcc_mean[19]
mfcc20_var=mfcc_var[19]
feature_array=np.array([length, chroma_stft_mean, chroma_stft_var,
rms_mean, rms_var,
        spectral_centroid_mean,
spectral_centroid_var,spectral_bandwidth_mean,spectral_bandwidth_v
ar,
        rolloff_mean,rolloff_var,zero_crossing_rate_mean,
zero_crossing_rate_var,

harmony_mean,harmony_var,perceptr_mean,perceptr_var,tempo,
        mfcc1_mean, mfcc1_var, mfcc2_mean,mfcc2_var,mfcc3_mean,
mfcc3_var,mfcc4_mean, mfcc4_var, mfcc5_mean, mfcc5_var,
mfcc6_mean,mfcc6_var,mfcc7_mean,mfcc7_var,mfcc8_mean,
mfcc8_var, mfcc9_mean, mfcc9_var,mfcc10_mean,mfcc10_var,
mfcc11_mean,mfcc11_var,
mfcc12_mean,mfcc12_var,mfcc13_mean,
        mfcc13_var, mfcc14_mean, mfcc14_var, mfcc15_mean,
mfcc15_var,
        mfcc16_mean, mfcc16_var, mfcc17_mean, mfcc17_var,
mfcc18_mean,
        mfcc18_var,mfcc19_mean, mfcc19_var, mfcc20_mean,
mfcc20_var])

```

```

song_test=feature_array.reshape(1,-1)
Song = pd.DataFrame(song_test, columns = ['length',
'chroma_stft_mean', 'chroma_stft_var', 'rms_mean', 'rms_var',
'spectral_centroid_mean', 'spectral_centroid_var',
'spectral_bandwidth_mean', 'spectral_bandwidth_var',
'rolloff_mean',
'rolloff_var', 'zero_crossing_rate_mean', 'zero_crossing_rate_var',
'harmony_mean', 'harmony_var', 'perceptr_mean', 'perceptr_var',
'tempo',
'mfcc1_mean', 'mfcc1_var', 'mfcc2_mean', 'mfcc2_var',
'mfcc3_mean',
'mfcc3_var', 'mfcc4_mean', 'mfcc4_var', 'mfcc5_mean',
'mfcc5_var',
'mfcc6_mean', 'mfcc6_var', 'mfcc7_mean', 'mfcc7_var',
'mfcc8_mean',
'mfcc8_var', 'mfcc9_mean', 'mfcc9_var', 'mfcc10_mean',
'mfcc10_var',
'mfcc11_mean', 'mfcc11_var', 'mfcc12_mean', 'mfcc12_var',
'mfcc13_mean',
'mfcc13_var', 'mfcc14_mean', 'mfcc14_var', 'mfcc15_mean',
'mfcc15_var',
'mfcc16_mean', 'mfcc16_var', 'mfcc17_mean', 'mfcc17_var',
'mfcc18_mean',
'mfcc18_var', 'mfcc19_mean', 'mfcc19_var', 'mfcc20_mean',
'mfcc20_var'])
pred=model.predict(Song)
predict(model,feature_array , 9)
price = pred[0]
fig = plt.figure(figsize =(10, 5))
plt.bar(lb, price[0:10])
plt.show()
return ipd.Audio(song)

```