# MUSIC GENRE CLASSIFICATION

Done by

**Ms. Maddali Sowmya**

**Ms. Pooja C**

Mentored by

**Ms. Gayathri V**

**Mr. Ragesh Verma**

In partial fulfillment of the requirement for

POSTGRADUATE DIPLOMA IN DATA SCIENCE AND ANALYTICS

NATIONAL INSTITUTE OF ELECTRONICS AND INFORMATION TECHNOLOGY [NIELIT] CHENNAI

(An Autonomous Scientific Society of Ministry of Electronics & Information Technology) Government of India

No. 25, Gandhi Mandapam Road, Chennai — 600025, Tamilnadu, India.

# DECLARATION

We hereby declare that the project work entitled

**MUSIC GENRE CLASSIFICATION**

Done for

**NATIONAL INSTITUTE OF ELECTRONICS**

**AND INFORMATION TECHNOLOGY [NIELIT] CHENNAI**

(An Autonomous Scientific Society of Ministry of Electronics & Information Technology)

Government of India

No.25, Gandhi Mandapam Road, Chennai- 600025, Tamilnadu, India.

**Under the Guidance of**

Ms. Gayathri

Mr. Ragesh

| | |
|---|---|
| **NAME OF THE STUDENT** | **Place:** Chennai |
| Ms. Maddali Sowmya | **Date:**10/03/2022 |
| Ms. Pooja C | |

# ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our guide Ms. Gayathri and Mr. Ragesh for the support for the project work, for their patience, motivation and immense knowledge.

Sincere thanks to our Dr. Sanjeev Kumar Jha, Joint Director (Tech.), NIELIT Chennai, for helping us to do this course & the project during the COVID19 Pandemic situation & also for giving us many supports and valuable guidance during our study.

A special thanks to our lecturers Ms. Lakshmipriya/Mr. Vignesh/Ms. Jayakodi/Mr. Saran/Ms. Gayatrhri for constant support, guidance, help and encouragement to learn in better way. They gave us the freedom to work freely with the topic by providing appropriate direction and materials, which was an unavoidable place for us to improve our knowledge and abilities from the start of the project to the end.

Our heartfelt gratitude and thanks to our parents for always helping us and supporting our education.

Finally heartfelt gratitude and thanks to our friends for constantly being there by our side all the time and believing in our ability to pursue this course and project.

# PROJECT SYNOPSIS

**Name of the Student:** Ms. Maddali Sowmya and Ms. Pooja C

**Course:** P.G. Diploma in Data Science & Analytics

**Institute/Company:** NIELIT, Chennai

**Guide Name:** Ms. Gayathri and Mr. Ragesh

**Project Title:** “Music Genre Prediction.”

**Abstract:** Music genres are helpful for grouping songs and artists to help listeners discover and explore new music. Because of the scope of this project, we decided that, rather than choosing a single model and maximizing our results, we would attempt various approaches to the problem, comparing the results of each approach to decide which model would be the most promising for this task for future researchers. In this project, we will describe our methodology and results for building a music genre classifier by training deep learning models on the GTZAN dataset (available on Kaggle). We attempt to compare and contrast models and approaches in order to determine which approach will work best for the task.

**Keywords:** Music genre, Classification, Deep Learning, Machine Learning.

# INDEX

# 1. AIM & OBJECTIVE

**AIM:** Today in 21st century, the usage of audio files has grown enormously. With large amount of audio files comes the need to classify the files to organise them without human intervention. Automatic Music Genre Classification (MGR) is a sub field of Music Information Retrieval (MIR). Algorithms use features of the sound files found in sound waves to classify the files. This project is aimed at developing such a solution to implement genre classification of audio files.

**OBJECTIVE:**

- GTZAN data set retrieval from Kaggle.
- Visualisation of data set using various methods
- Pre-processing the data and extracting features from it to transform the raw data into an understandable format
- Use of XGB classifier to classify the audio input files to its genre with high accuracy.

# 2. FIGURES

# 3. INTRODUCTION

Over the past decade, large collections of music are increasingly available on the various application platforms. Therefore, tasks such as music discovery, navigation and organisation have become progressively harder for humans without the help of automated systems. Extensive research effort has been invested in music information retrieval at the intersection of signal processing, music modelling, and machine learning.

Music informative retrieval has assumed lot of significance in the recent past owling to wide business applications. These includes recommender systems, track separation and instrument recognition, automatic music transcription, automatic categorisation / genre classification etc.

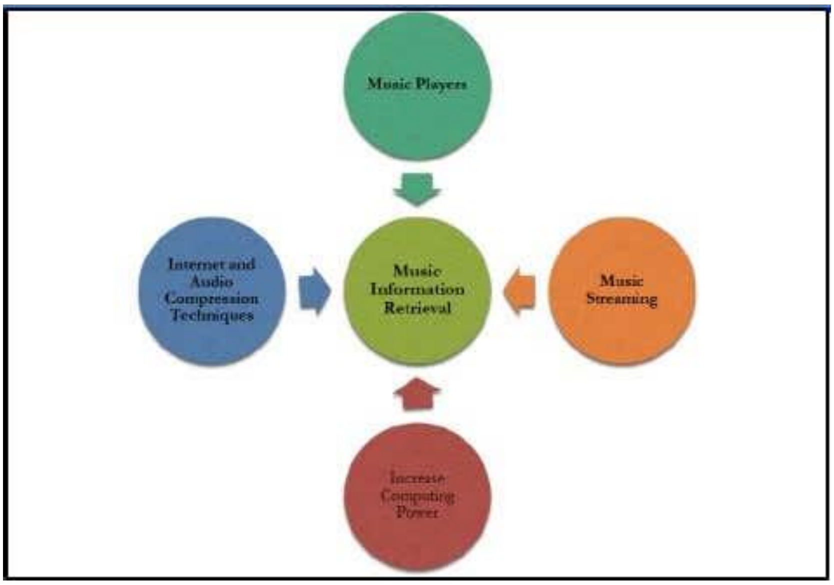

Music Players
Internet and Audio Compression Techniques
Music Information Retrieval
Music Streaming
Increase Computing Power

*Fig.3.1. The technology dynamics that are driving the Musical Information Retrieval*

Consumption of music online via streaming has gained popularity as downloading and storing music files has become easier, large collection of albums are available on the cloud either as a free service or as a paid service. One of the key elements of music data management is to identify a particular audio file with respect to the genre it belongs to and store such large quantity of files grouped by genre for easier management. These days, online radio stations play songs to a particular user based on the genre preference. Many online streaming services recommend to play a specific song / an audio clip for a given user based on their browsing or search history on internet and these streaming services also come up with a concept called smart playlists based on music played by user or preferences. With such diverse application and a large volume of music data being used, music database management is inevitable and it is becoming a big data problem to solve these days.

## a. PROBLEM STATEMENT:

The question of which genre a music file belongs to, is a question of classification – a semantic problem. Music can be classified by its time of creation, geographical origin, topic or set of rules related to the sound. Some of these facts are often added to the files as metadata because they cannot be retrieved from the sound waves. Humans however classify music by the perception of the sound produced by the audio signal. Music genre is subjective from person to person and can be ambiguous. On top of that, a music file can be assigned to more than one genre and using more than one classification category. It is therefore questionable to speak in terms of “accuracy”, “hit” or “miss” if a song cannot be classified to a genre. Every accuracy value can therefore be regarded as some fuzzy approximation.

Music genre classification is achieved by learning the characteristics of collections of songs for which genres are already determined. This method is termed as supervised learning technique. Another approach is unsupervised learning, in this approach unlabelled songs are analysed by examining their characteristics. This project aims to visualise the music dataset given, pre-process the data, extract features and build a model to classify the music file to its genre accurately.

## b. DATASET:

In this project, we have taken GTZAN dataset from Kaggle. GTZAN dataset is composed of 1,000 30-second clips covering ten genres, with 100 clips per genre in .au format. GTZAN dataset primarily comprises of western musical genres such as Blues, Classical, Country, Disco, Hip Hop, Jazz, Metal, Pop, Reggae and Rock.

# 4. LITERATURE REVIEW

I. Paper [1], proposed the answer to meet the matter of blurry classification of Pop, Rock and Electronic genres. It seems that these genres don't seem to be arranged accurately by the above classifiers. The behaviour of the given classifier on the quality musical style dataset GTZAN genre collection that consists of 1000 songs and Free Music Archive (FMA) dataset is investigated. The proposed hybrid classifier is deployed on spark platform to demonstrate the scalability of the system. Here the input audio signal in divided into frames of constant frame-size, hopping thought the audio signal with a relentless hop-size, features are extracted on one frame at a time.

II. Paper [2], gives a novel method of musical style recognition using an ensemble of convolutional long short-term memory based neural networks and a transfer learning model. The neural network models are trained by various set of concerning spectral and rhythmic features compared to the transfer learning model was originally trained on the task of music tagging. Here the deep neural network models are trained on a group of extracted spectral and rhythmic features. Used transfer learning system to extract meaningful features from the songs. A multilayer computer model network is then trained on this transferred feature to predict the genres. Finally, the predictions of various models are combined employing a majority voting ensemble.

III. Paper [3], aims to get a low-computational and data budget model. It makes use of well-known architectures, different strategies for fine-tuning, initializations and optimizers to get that matches better within the musical style classification. Also, it makes use of multi-frame approach with a median stage to analyse intimately almost the total full song. Here the aim is to find out from a source data distribution a well performing model on a special target data distribution. The two common practices and therefore the ones that are applied are: Using the network as feature extractor. That is removing the last fully-connected layer and treats the network as a feature extractor. Once we have extracted all the features at the top we can include a classifier like SVM or a SoftMax classifier for the new dataset. Fine- tuning the network: This strategy is based on not only replace the classifier layer of the network, but also retrain part or the whole network. Through back propagation we can modify the weights of the pre-trained model to adapt the model to the new data distribution. Sometimes it’s preferable to keep the first layers of the network fixed (or freezed) to avoid overfitting, and only fine-tune the deeper part. This is motivated because the lower layers of the networks capture generic features that are similar to many tasks while the higher layers contain features that are task and dataset oriented as demonstrated in.

IV. In paper [4], comparison between two classes of model is implemented. The first is a deep learning approach wherein a CNN model is trained end-to-end, to predict the genre label of an audio signal, solely using spectrogram. The second approach uses both hand crafted features from the time domain and frequency domain. Thus, to these process features that serves the most in classification of task is established. Here two different approaches are used to solve the problem. The first approach generates a spectrogram of the audio signal treating it as an image. A CNN based image classification model, namely VGG-16 is also skilled on these images to estimate the music genre which is based on the spectrogram image. The second approach was to extract the time domain and frequency domain features from the audio signals followed by training traditional machine learning classifier based on the extracted features. XGBoost was used as a classifier to report the most important features. Also, assembling the CNN and XGBoost model was carried out and it turned out to be beneficial.

V. In paper [5], convolution neural network deep learning approach is used in order to train and classify the system. Here convolution neural network is used for the training and classification purpose. Feature Extraction is the most important task for audio analysis and hence Mel Frequency Cepstral Coefficient (MFCC) is used as a feature vector for sound sample. The proposed system classifies music into various genres by retrieving the feature vector. This feature is obtained by taking the Fourier transforms of the signal, then taking the logarithmic of the power values and then taking the cosine transforms. The detailed explanation will be done in the forthcoming sessions. These extracted features then act as the inputs to the neuros for training. For our work, we analyse music from ten various genres.

# 5. CLASSIFICATION METHODOLOGY

The GTZAN dataset we used for this project had pre-defined classes for each audio file. Hence, we used CNN and XGB classifier which is supervised-learning algorithm used for regression and classification of large datasets.

The methodology is dived into four steps:

- Data Collection
- Feature Extraction
- Analysis and Modelling
- Evaluate and improve

The overall methodology followed for classification is summarised as shown in the picture below.

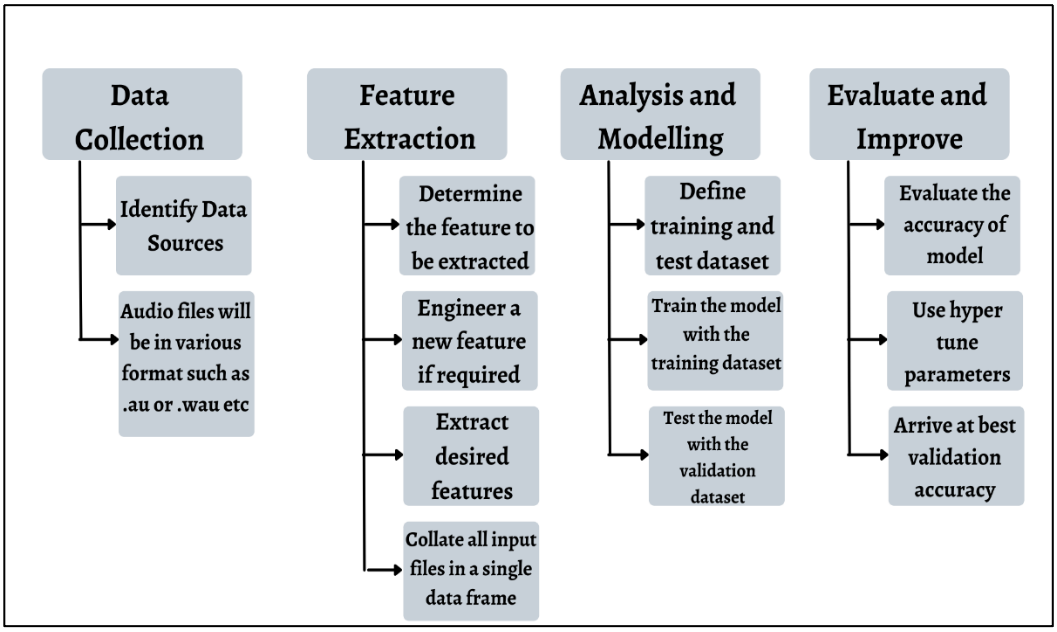

Data Collection
Identify Data Sources
Audio files will be in various format such as .au or .wau etc
Feature Extraction
Determine the feature to be extracted
Engineer a new feature if required
Extract desired features
Collate all input files in a single data frame
Analysis and Modelling
Define training and test dataset
Train the model with the training dataset
Test the model with the validation dataset
Evaluate and Improve
Evaluate the accuracy of model
Use hyper tune parameters
Arrive at best validation accuracy

*Fig. 5.1 Summary of Classification Methodology*

# 6. DATA VISUALIZATION

Data visualization is the representation of data through use of common graphics, such as charts, plots, infographics, and even animations. These visual displays of information communicate complex data relationships and data-driven insights in a way that is easy to understand.

No matter what business or career is chosen, data visualization can help by delivering data in the most efficient way possible. As one of the essential steps in the business intelligence process, data visualization takes the raw data, models it, and delivers the data so that conclusions can be reached. In advanced analytics, data scientists are creating machine learning algorithms to better compile essential data into visualizations that are easier to understand and interpret.

Specifically, data visualization uses visual data to communicate information in a manner that is universal, fast, and effective. This practice can help companies identify which areas need to be improved, which factors affect customer satisfaction and dissatisfaction, and what to do with specific products (where should they go and who should they be sold to). Visualized data gives stakeholders, business owners, and decision-makers a better prediction of sales volumes and future growth.

There are many types of data visualisation techniques, few of them are- Tables, Pie charts and stacked bar charts, Line graphs and area charts, Histograms, Scatter plots, Heat maps, Tree maps. In our project we have used Raw Plot Wave, Spectrogram Plot and Zero Crossing Rate to visualize the audio.

- ***Raw Plot wave:*** This type of visualization is used to visualize our audio file and its amplitude in a waveform format using "librosa.display.waveshow" function. This function constructs a plot which adaptively switches between a raw samples-based view of the signal and an amplitude-envelope view of the signal depending on the time extent of the plot's viewport. When visualizing stereo waveforms, the amplitude envelope will be generated so that the upper limits derive from the left channel, and the lower limits derive from the right channel, which can produce a vertically asymmetric plot. More specifically, when the plot spans a time interval of less than max_points / sampling rate (by default, 1/2 second), the samples-based view is used, and otherwise a downsampled amplitude envelope is used. This is done to limit the complexity of the visual elements to guarantee an efficient, visually interpretable plot. In our project we have taken one audio file from the GTZAN dataset and using librosa.display.waveshow function we have plotted the raw wave plot of the audio file. The figure below shows the raw wave plot, which depicts amplitude envelope of a waveform.

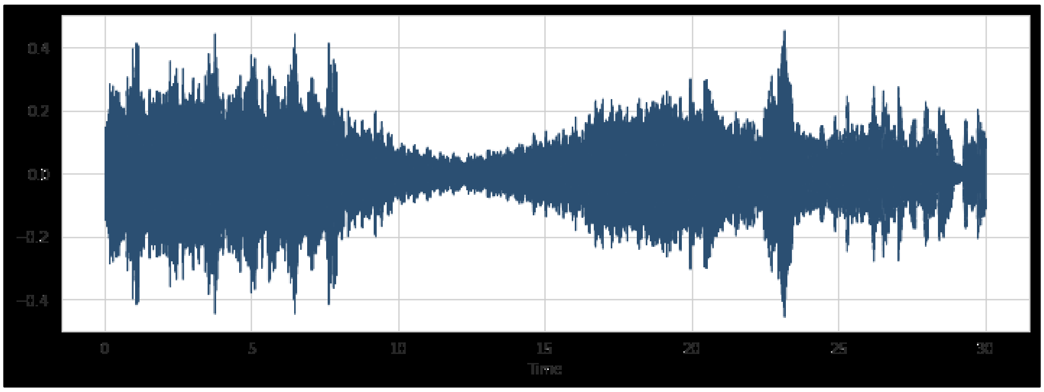


*Fig.6.1. Raw wave plot in time domain*

- ***Spectrogram Plot:*** A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time. When applied to an audio signal, spectrograms are sometimes called sonographs, voiceprints, or voicegrams. When the data are represented in a 3D plot they may be called waterfall displays. A spectrogram is like a photograph or image of a signal. It plots time in Y-axis and frequencies in X-axis. It also conveys the signal strength using the colors – brighter the color the higher the energy of the signal. The spectrogram explains how the signal strength is distributed in every frequency found in the signal. These are actually created using Short-time Fourier Transform (STFT). It helps us to do a time-varying analysis of the signal provided. The main concept is that we divide the audio signal into small pieces and then that audio signal is plotted on the graph against time. The python module matplotlib.pyplot provides the specgram() method which takes signal as an input and plots the spectrogram. The specgram() method uses Fast Fourier Transform(FFT) to get the frequencies present in the signal. This method takes several parameters that customizes the spectrogram based on a given signal. The figure below represents the spectrogram plot of the audio file where x-axis is time and y-axis is frequency.

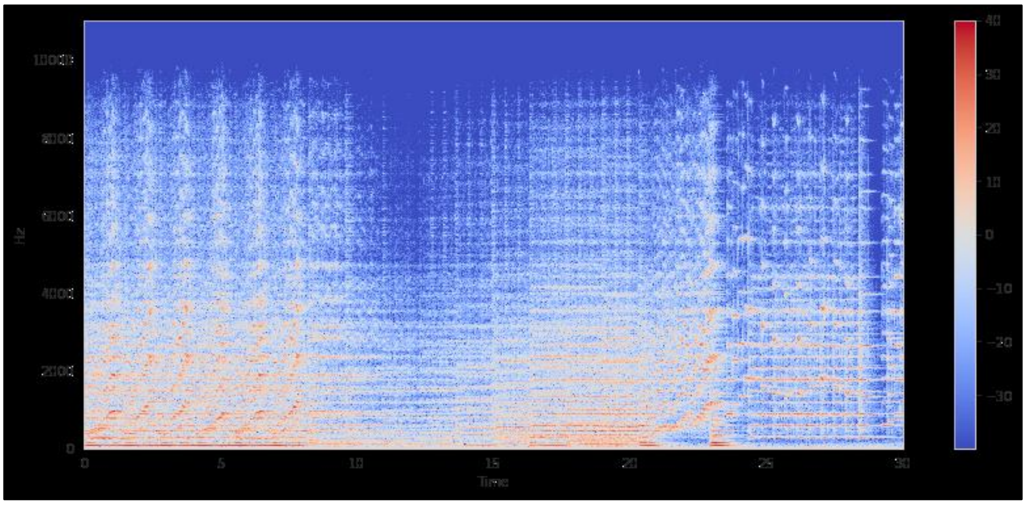


*Fig.6.2. Spectrogram plot of audio file*

The vertical axis represents frequencies (from 0 to 10kHz), and the horizontal axis represents the time of the clip. Next, we have converted the frequency axis to logarithm, because we see that all actions (in red) are taking place at the bottom of the spectrum.

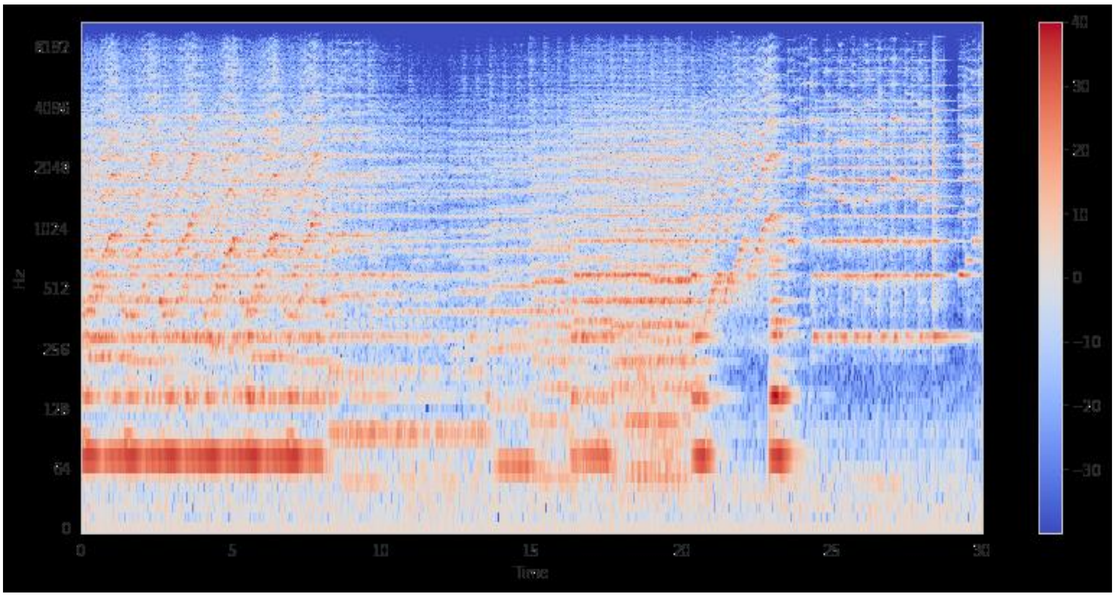


*Fig.6.3. Spectrogram of audio file in time and logarithmic domain.*

- ***Zero-Crossing Rate:*** A zero-crossing is an instantaneous point at which the sign of a mathematical function changes (e.g. from positive to negative). It is represented by an intercept of the axis (zero value) in the graph of the function. The zero-crossing rate (ZCR) is the rate at which a signal transitions from positive to zero to negative or negative to zero to positive. Its value has been extensively used in both speech recognition and music information retrieval for classifying percussive sounds.

ZCR is defined as:

$$zcr = \frac{1}{T-1}\sum_{t=1}^{T-1} 1_{\mathbb{R}_{<0}}(s_t s_{t-1})$$

The zero-crossing rate can be utilized as a basic pitch detection algorithm for monophonic tonal signals. Voice activity detection (VAD), which determines whether or not human speech is present in an audio segment, also makes use of zero-crossing rates.

A zero crossing is said to occur if successive samples have different algebraic signs. The rate at which zero crossings occur is a simple measure of the frequency content of a signal. Zero-crossing rate is a measure of number of times in a given time interval/frame that the amplitude of the speech signals passes through a value of zero The below figure shows the zero-crossing graph of audio signal.

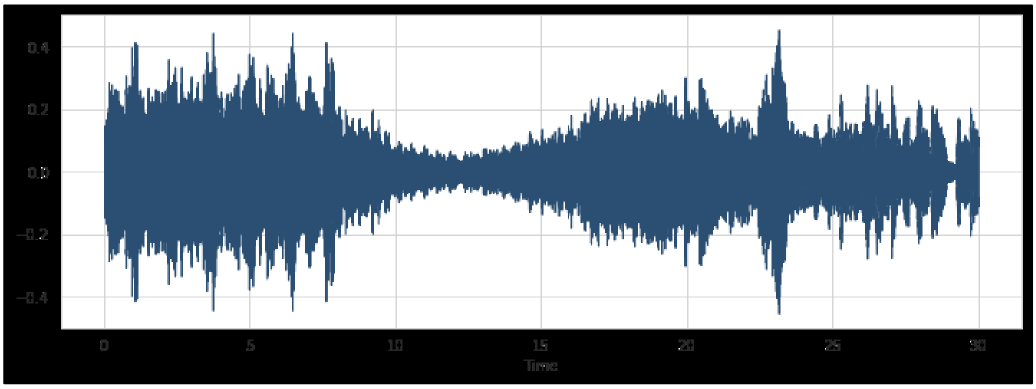


*Fig.6.4. Zero crossing graph of audio signal*

# 7. FEATURE EXTRACTION OF GTZAN DATASET

- ***Mel-Frequency Cepstral Coefficients (MFCC)***: Human perception of the frequency content of sounds does not follow a linear scale but uses a logarithmic distribution. MFCCs are based on the spectral information of sound, but are modelled to capture the perceptually relevant parts of auditory spectrum. The sequence of processing is as follows:
    - Window the data (e.g. with a Hamming window);
    - Calculate the magnitude of the FFT;
    - Convert the FFT data into filter bank outputs;
    - Calculate the log base 10;
    - Calculate the cosine transform.

The filter bank is what makes MFCCs unique. It is constructed using 13 linearly spaced filters and 27 log spaced filters, following a common model for human auditory perception. The distance between the centre frequencies of the linearly spaced filters is 133,33 Hz; the log-spaced filters are separated by a factor of 1.071 in frequency. The final cosine transform (step 5) is applied to reduce the dimensionality of the output, typically to the 12 most important coefficients. Additionally, the power of the signal for each frame is calculated, resulting in a feature vector of d = 13. MFCCs are commonly used in speech recognition systems, and seem to capture the perceptually relevant part of the spectrum better than other techniques. They have successfully been applied to the content-based retrieval of audio samples and also used in music genre recognition systems. The MFCC plot is harder to interpret visually than the spectrogram, but has been found to yield better results in computer sound analysis.

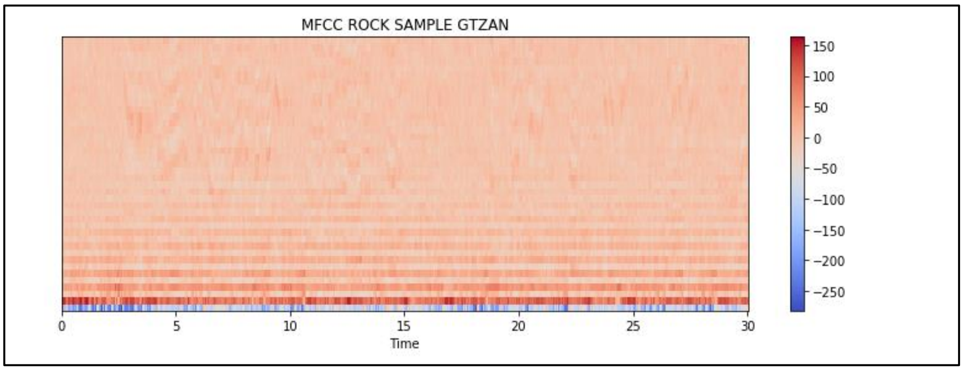


*Fig.7.1 Sample plot of MFCC for a rock music file from GTZAN dataset*

- ***Mel Spectrogram***: Mel spectrogram represents an acoustic time-frequency representation of a sound: the power spectral density P(f, t).It is sampled into a number of points around equally spaced times ti and frequencies fj (on a Mel frequency scale). The Mel frequency scale is defined as:

mel = 2595 * log10 (1 + hertz / 700),

and its inverse is:

hertz = 700 * (10.0mel / 2595.0 - 1).

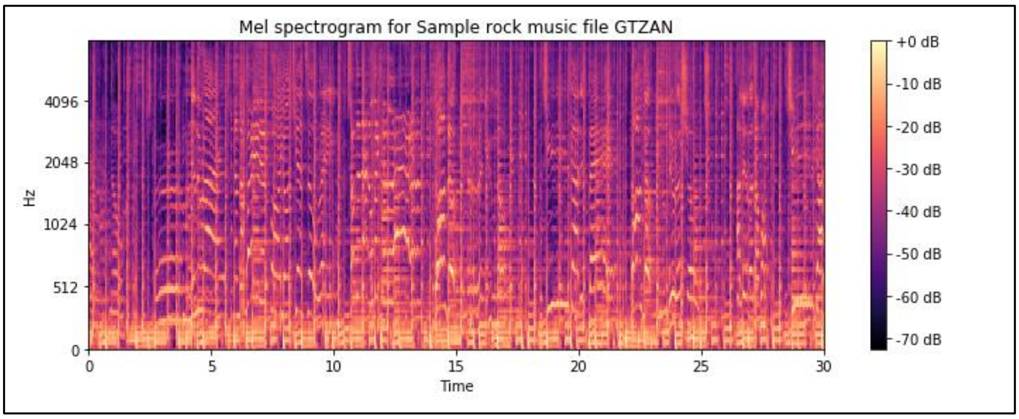


*Fig.7.2. Mel spectrogram for sample rock music file GTZAN*

- ***RMSE:*** The energy [10] of a signal corresponds to the total magnitude of the signal. For audio signals, that roughly corresponds to how loud the signal is. The energy in a signal is defined as

$$\sum_n |x(n)|^2$$

The root-mean-square energy (RMSE) in a signal is defined as

$$\sqrt{\frac{1}{N}\sum_n |x(n)|^2}$$

- ***Chromagram features:*** In the music context, the term chroma feature or chromagram closely relates to the twelve different pitch classes. Chroma-based features, which are also referred to pitch class profiles, are a powerful tool for analyzing music whose pitches can be meaningfully categorized (often into twelve categories) and whose tuning approximates to the equal-tempered scale. One main property of chroma features is that they capture harmonic and melodic characteristics of music, while being robust to changes in timbre and instrumentation. Identifying pitches that differ by an octave, chroma features show a high degree of robustness to variations in timbre and closely correlate to the musical aspect of harmony. This is the reason why chroma features are a well-established tool for processing and analyzing music data. For example, every chord recognition procedure relies on some kind of chroma representation. Also, chroma features have become the de facto standard for tasks such as music alignment and synchronization as well as audio structure analysis. Finally, chroma features have turned out to be a powerful mid-level feature representation in content-based audio retrieval such as cover song identification or audio matching. In the current project we used Chroma variant “Chroma Energy Normalized”(Chroma CEN).

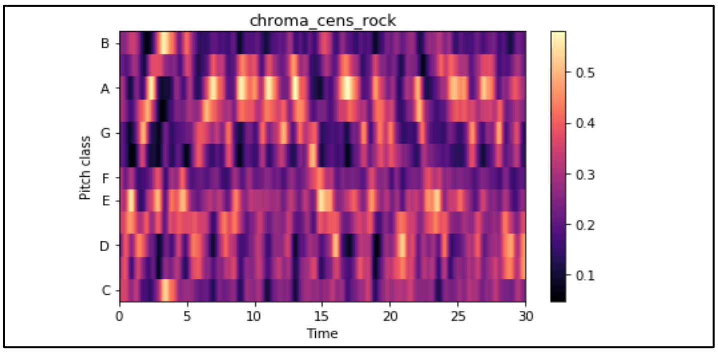


*Fig.7.3. Sample plot of Chroma Normalized feature for Rock file form GTZAN*

- ***Spectral Centroid:*** The spectral centroid is a measure used in digital signal processing to characterize a spectrum. It indicates where the "center of mass" of the spectrum is located. Perceptually, it has a robust connection with the impression of "brightness" of a sound. It is calculated as the weighted mean of the frequencies present in the signal, determined using a Fourier transform, with their magnitudes as the weights:

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

where x(n) represents the weighted frequency value, or magnitude, of bin number n, and f(n) represents the center frequency of that bin, because the spectral centroid is a good predictor of the "brightness" of a sound, it is widely used in digital audio and music processing as an automatic measure of musical timbre.

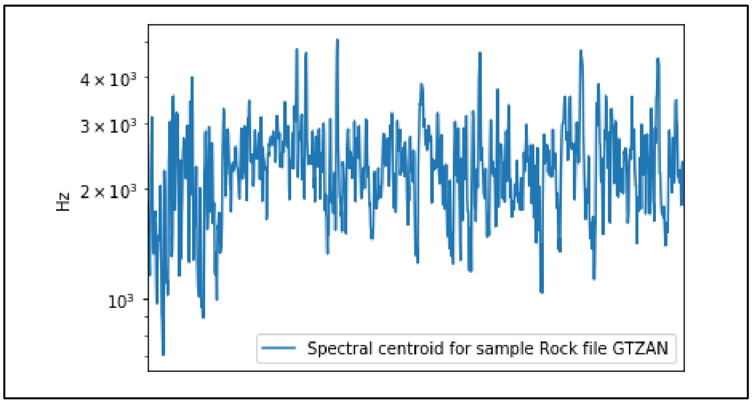


*Fig.7.4. Spectral Centroid Plot for Sample rock file from GTZAN dataset*

- ***Spectral Contrast:*** Octave-based Spectral Contrast [12] introduced by Jiang et al. considers the spectral peak, spectral valley and their difference in each sub-band. For most music, the strong spectral peaks roughly correspond with harmonic components; while non-harmonic components, or noises, often appear at spectral valleys. Thus, Spectral Contrast feature could roughly reflect the relative distribution of the harmonic and non-harmonic components in the spectrum.

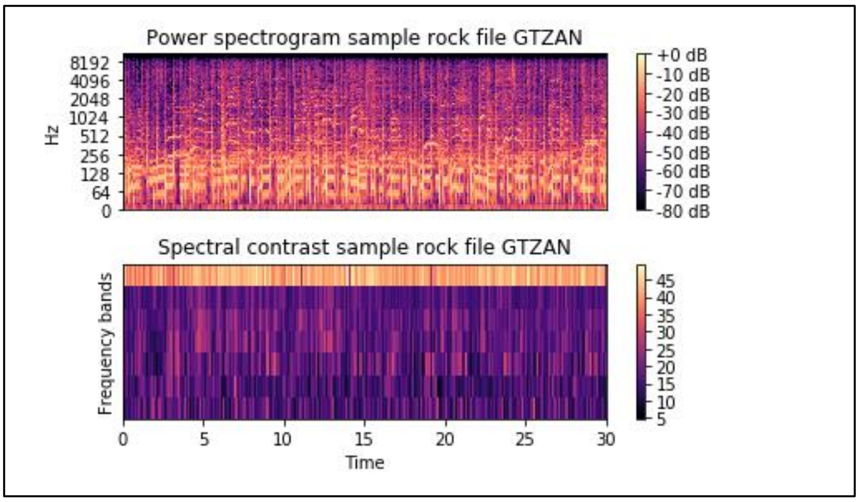


*Fig.7.5. Power Spectrogram and Spectral Contrast of sample rock file from GTZAN dataset*

- ***Tonal Centroid***: Tonal Centroid introduced by Harte et al. [13] maps a Chroma gram onto a six-dimensional Hypertorus structure. The resulting representation wraps around the surface of Hypertorus, and can be visualized as a set of three circles of harmonic pitch intervals: fifths, major thirds and minor thirds. Tonal Centroids are efficient in detecting the changes in harmonic contents.

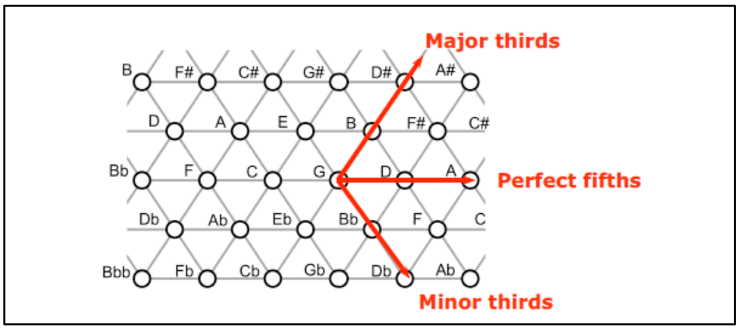


*Fig.7.6. Tonal Centroid plot for sample rock file from GTZAN dataset*

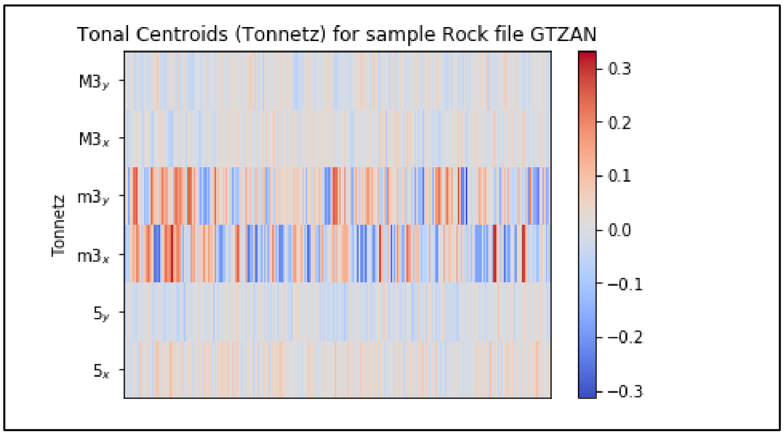


*Fig.7.7. Tonnetz for sample rock file GTZAN.*

# 8. ALGORITHMS AND LIBRARIES USED

- ***EDA AND DATA PROCESSING:*** Exploratory Data Analysis (EDA) is an essential step in any research analysis. The primary aim with exploratory analysis is to examine the data for distribution, outliers and anomalies to direct specific testing of your hypothesis. It also provides tools for hypothesis generation by visualising and understanding the data usually through graphical representation. EDA aims to assist the natural pattern recognition of the analyst. EDA is fundamental early step after data collection and data pre-processing, where the data is simply visualised, plotted and manipulated, without any assumptions, in order to help assessing the quality of the data and building models. The reason for the heavy reliance on graphics is that by its very nature the main role of EDA is to explore, and graphics gives the analysts unparalleled power to do so, while being ready to gain insight into the data. There are many ways to categorise the many EDA techniques. Here in our project, we have used correlations between mean variables, which is multi-variate quantitative EDA. Correlation evaluates the direction as well as strength of a relationship between continuous variables. Correlation coefficient can range from -1 to +1, which signifies strong negative to strong positive relation between the variables. Correlation=0, suggests that two variables are independent of each other. A positive correlation point towards the positive relationship between two variables, which means if one variable a changes the second variable also changes is same direction (increase or decrease).

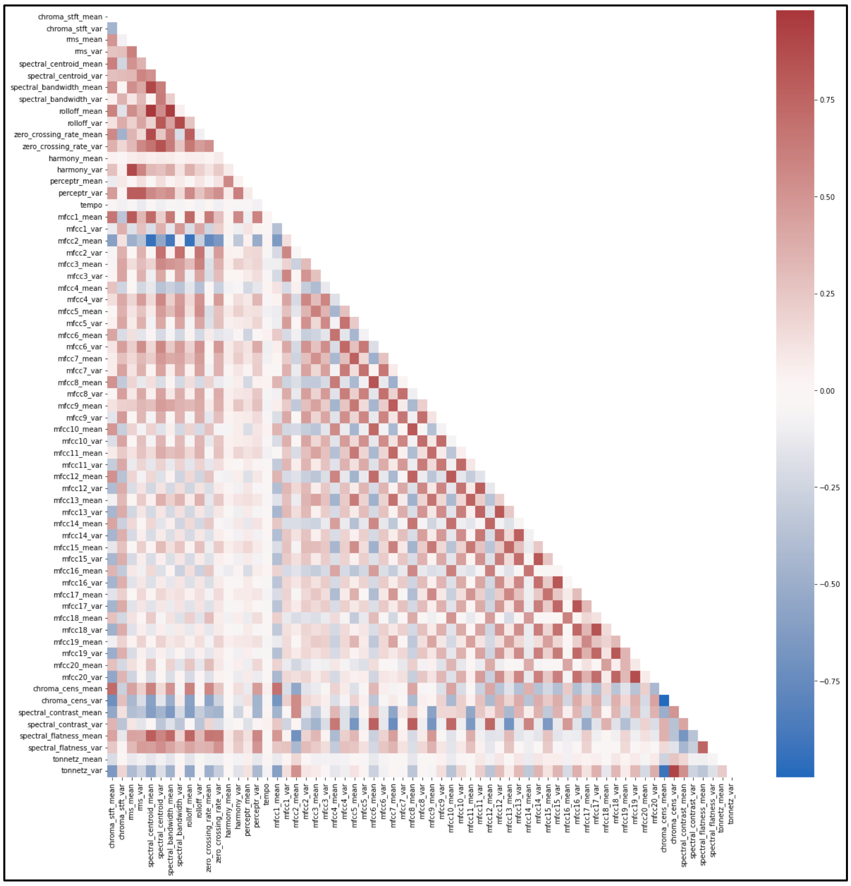


*Fig.8.1. Correlation between mean variables*

Data pre-processing consists of a series of steps to transform raw data derived from data extraction into a “clean” and “tidy” dataset prior to statistical analysis.

Several distinct steps are involved in pre-processing data. Here are the general steps taken to pre-process data:

- Data “cleaning”—This step deals with missing data, noise, outliers, and duplicate or incorrect records while minimizing introduction of bias into the database.
- “Data integration”—Extracted raw data can come from heterogeneous sources or be in separate datasets. This step reorganizes the various raw datasets into a single dataset that contain all the information required for the desired statistical analyses.

- "Data transformation"—This step translates and/or scales variables stored in a variety of formats or units in the raw data into formats or units that are more useful for the statistical methods that the researcher wants to use.
- "Data reduction"—After the dataset has been integrated and transformed, this step removes redundant records and variables, as well as reorganizes the data in an efficient and "tidy" manner for analysis.

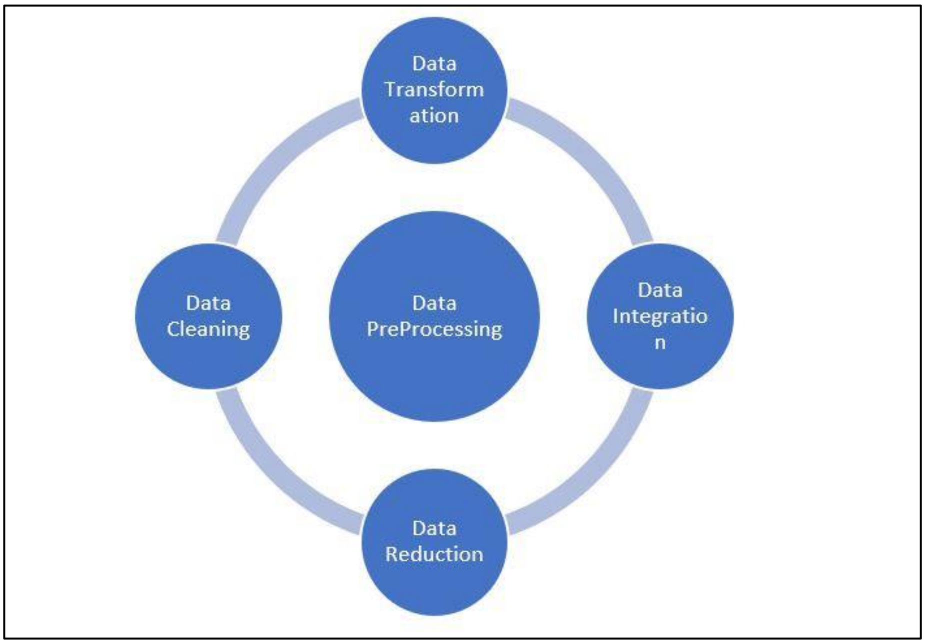
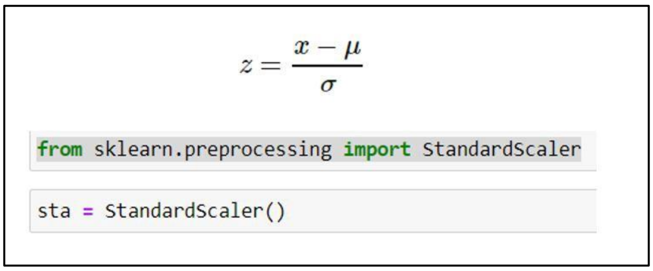


*Fig.8.2. Data Pre-processing techniques*

Pre-processing is sometimes iterative and may involve repeating this series of steps until the data are satisfactorily organized for the purpose of statistical analysis. During pre-processing, one needs to take care not to accidentally introduce bias by modifying the dataset in ways that will impact the outcome of statistical analyses. In our project, we use data transformation technique which aims to transform the data values into a format, scale or unit that is more suitable for analysis (e.g. log transform for linear regression modelling). There are many possible options in data transformation but we have used "StandardScalar" technique of normalisation in our project. StandardScaler standardizes a feature by subtracting the mean and then scaling to unit variance. Unit variance means dividing all the values by the standard deviation. Standardization can be helpful in cases where the data follows a Gaussian distribution (or Normal distribution). However, this does not have to be necessarily true. Also, unlike normalization, standardization does not have a bounding range. So, even if you have outliers in your data, they will not be affected by standardization.

where μ is the mean (average) and σ is the standard deviation from the mean; standard scores (also called z scores) of the samples are calculated as follows



$$z = \frac{x - \mu}{\sigma}$$

```
from sklearn.preprocessing import StandardScaler
```

```
sta = StandardScaler()
```

*Fig.8.3. StandardScaler formula and import code*

StandardScaler results in a distribution with a standard deviation equal to 1. The variance is equal to 1 also, because variance = standard deviation squared. And 1 squared = 1. StandardScaler makes the mean of the distribution 0. About 68% of the values will lie be between -1 and 1. Deep learning algorithms often call for zero mean and unit variance. Regression-type algorithms also benefit from normally distributed data with small sample sizes. It does distort the relative distances between the feature values.

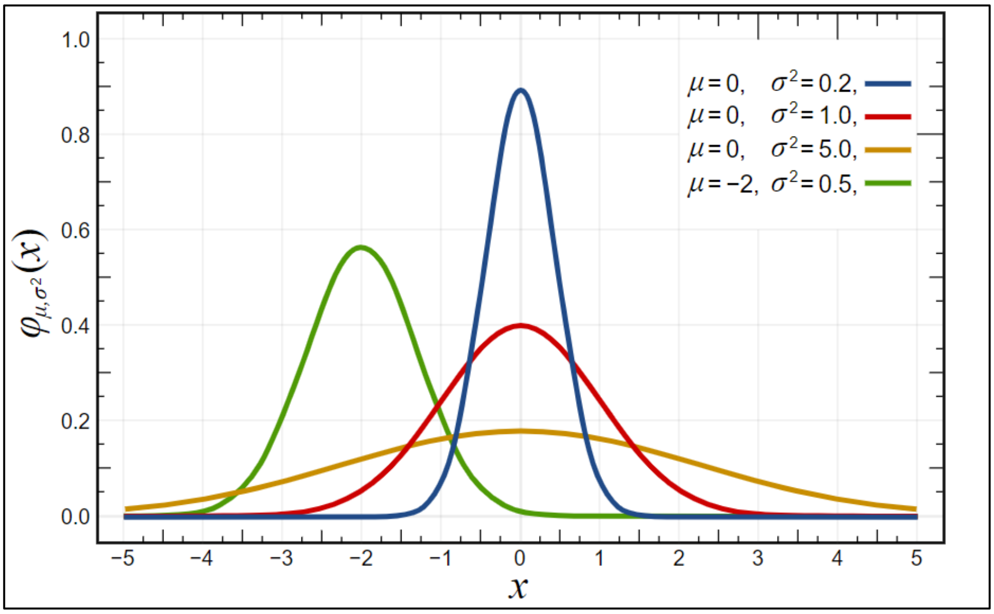


*Fig.8.4. Bell Curve or Gaussian Distribution or Normal Distribution*

- ***XGB CLASSIFIER:*** eXtreme Gradient Boosting (XGBoost) is a scalable and improved version of the gradient boosting algorithm designed for efficacy, computational speed and model performance. It is an open-source library and a part of the Distribution Machine Learning community. XGB is a perfect blend of software and hardware capabilities designed to enhance existing boosting techniques with accuracy in the shortest amount of time.

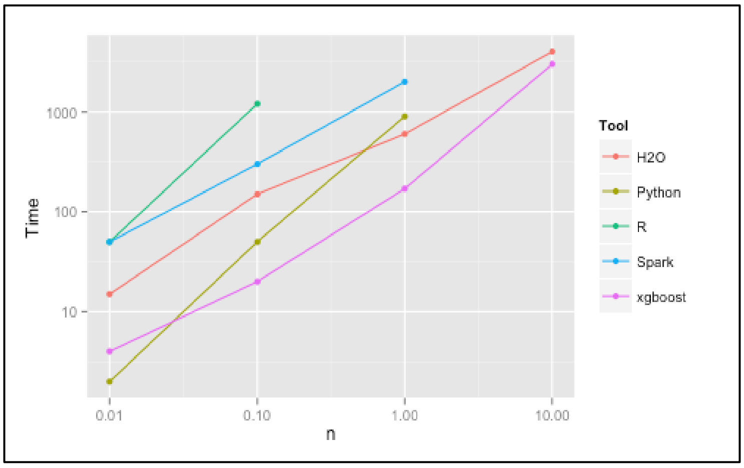


*Fig.8.5. Benchmark performance of XGBoost*

This takes us to the question of what is the history of boosting? Let’s dive deep into it.

- A quick flashback to Boosting: Boosting generally means increasing performance. In ML, boosting is a sequential ensemble learning technique to convert a weak hypothesis or weak learners into strong learners to increase the accuracy of the model.

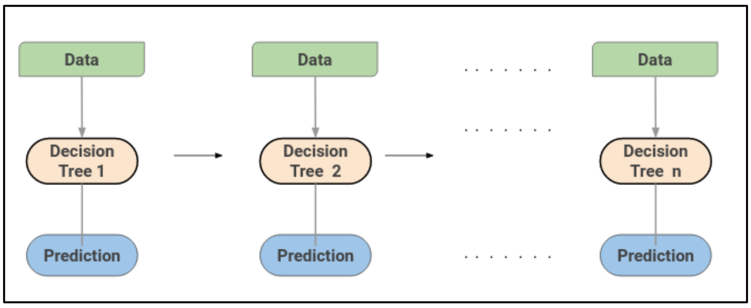


*Fig.8.6. Boosting Algorithms in Machine Learning*

- Ensemble Learning: Ensemble learning is a process in which decisions from multiple machine learning models are combined to reduce errors and improve predictions when compared to single ML model. Then maximum voting technique is used on aggregate decisions (or predictions in machine learning jargon) to deduce the final prediction. The image below shows the clear difference between single ML model and ensemble ML model.

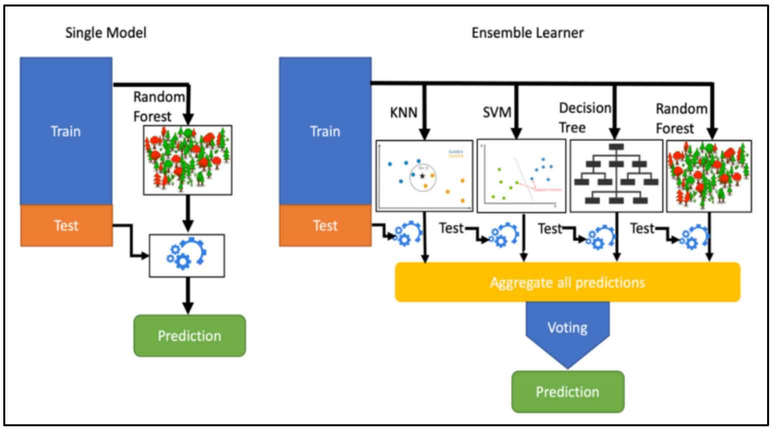


*Fig.8.7. Single ML model vs Ensemble ML model*

- Working of Boosting algorithm: The boosting algorithm creates new weak learners (models) and sequentially combines their predictions to improve the overall performance of the model. For any incorrect prediction, larger weights are assigned to misclassified samples and lower ones to samples that are correctly classified. Weak learner models that perform better have higher weights in the final ensemble model. Boosting never changes the previous predictor and only corrects the next predictor by learning from mistakes. The first implementation of boosting was named AdaBoost (Adaptive Boosting).

$$F_i(x) = F_{i-1}(x) + f_i(x)$$

Where, F(i) is current model, F(i-1) is previous model and f(i) represents weak model

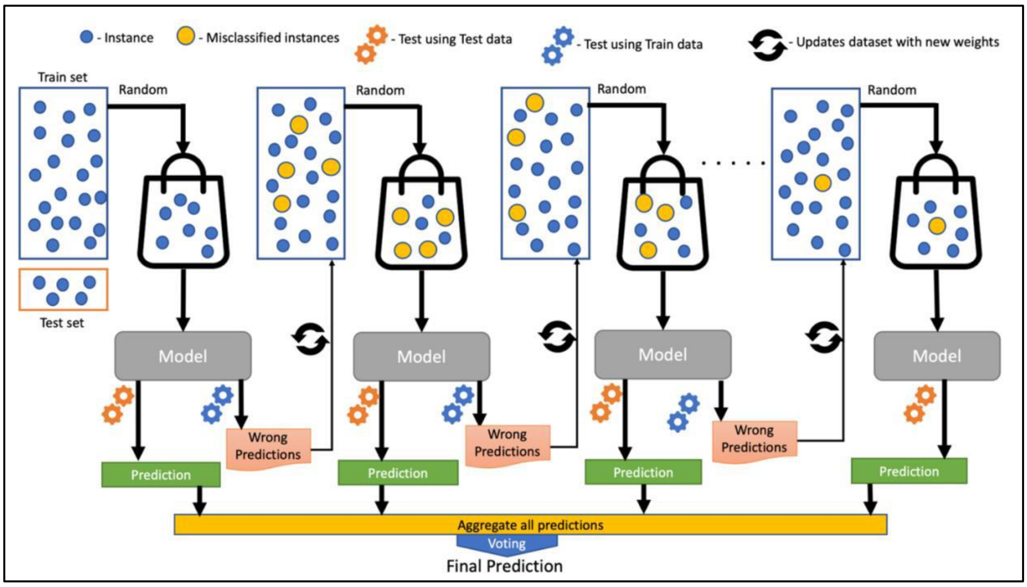


*Fig.8.8. Internal working of boosting algorithm*

- Gradient Boosting: Gradient boosting is a special case of boosting algorithm where errors are minimized by a gradient descent algorithm and produce a model in the form of weak prediction models e.g. decision trees. The major difference between boosting and gradient boosting is how both the algorithms update model (weak learners) from wrong predictions. Gradient boosting adjusts weights by the use of

$$w = w - \eta \nabla w$$

$$\nabla w = \frac{\partial L}{\partial w} \; where\ L\ is\ loss$$

gradient (a direction in the loss function) using an algorithm called Gradient Descent, which iteratively optimizes the loss of the model by updating weights. Loss normally means the difference between the predicted value and actual value. For regression algorithms, we use MSE (Mean Squared Error) loss while for classification problems we use logarithmic loss.

Where, w represents the weight vector, η is the learning rate.

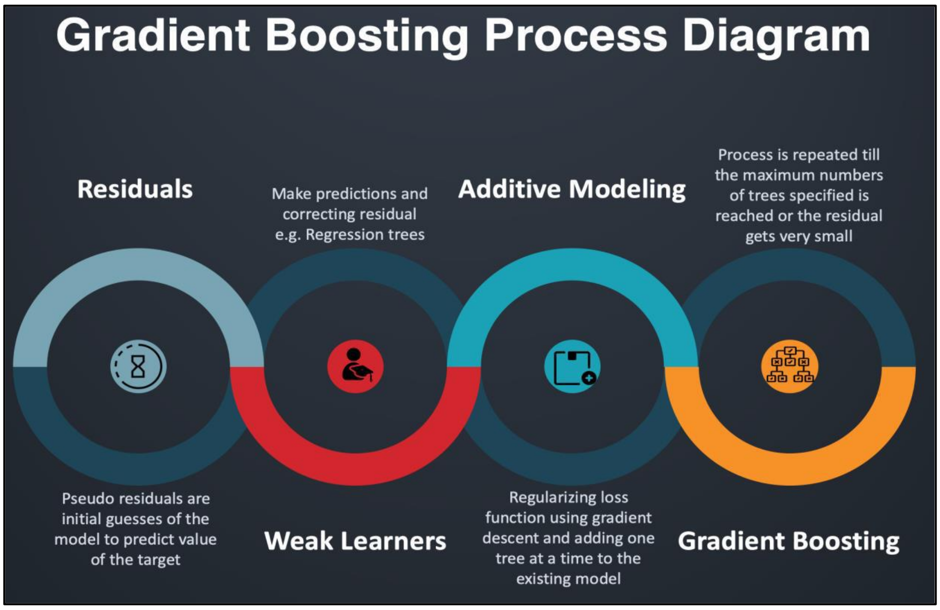

Gradient Boosting Process Diagram
Residuals
Make predictions and correcting residual e.g. Regression trees
Additive Modeling
Process is repeated till the maximum numbers of trees specified is reached or the residual gets very small
Pseudo residuals are initial guesses of the model to predict value of the target
Weak Learners
Regularizing loss function using gradient descent and adding one tree at a time to the existing model
Gradient Boosting

*Fig.8.9. Process flow of gradient boosting*

Gradient boosting uses Additive Modeling in which a new decision tree is added one at a time to a model that minimizes the loss using gradient descent. Existing trees in the model remain untouched and thus slow down the rate of overfitting. The output of the new tree is combined with the output of existing trees until the loss is minimized below a threshold or specified limit of trees is reached. Additive Modeling in mathematics is a breakdown of a function into the addition of N subfunctions. In statistical terms, it can be thought of as a regression model in which response y is the arithmetic sum of individual effects of predictor variables x.

- **XGBOOST in action:** XGBOOST is used in various different Kaggle competitions and etc for its robust features. Here are the features with details and how they are incorporated in XGBoost to make it robust.

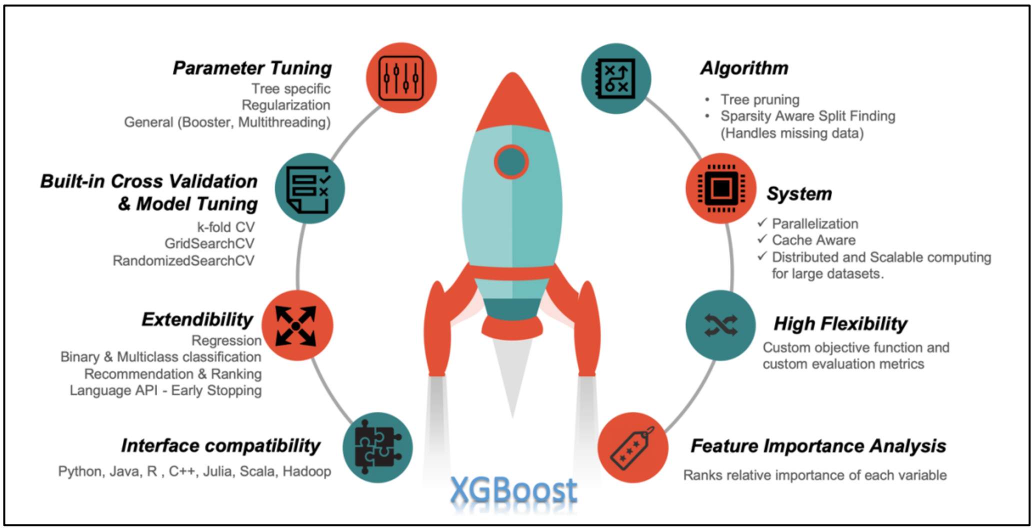

Parameter Tuning
Tree specific
Regularization
General (Booster, Multithreading)
Built-in Cross Validation & Model Tuning
k-fold CV
GridSearchCV
RandomizedSearchCV
Extendibility
Regression
Binary & Multiclass classification
Recommendation & Ranking
Language API - Early Stopping
Interface compatibility
Python, Java, R , C++, Julia, Scala, Hadoop
Algorithm
• Tree pruning
• Sparsity Aware Split Finding (Handles missing data)
System
✓ Parallelization
✓ Cache Aware
✓ Distributed and Scalable computing for large datasets.
High Flexibility
Custom objective function and custom evaluation metrics
Feature Importance Analysis
Ranks relative importance of each variable
XGBoost

*Fig.8.10. Features of XGBOOST*

In our project, we have first divided our dataset into training and test dataset. In the initial process of using XGBOOST we have done model fitting and recursive feature elimination. After eliminating the features, we are again fitting the XGBOOST-classifier to the modified dataset to classify the dataset into its genre accurately. Using the confusion matrix of the test dataset we were able to visualize important predictive analytics like recall, specificity, accuracy, and precision.

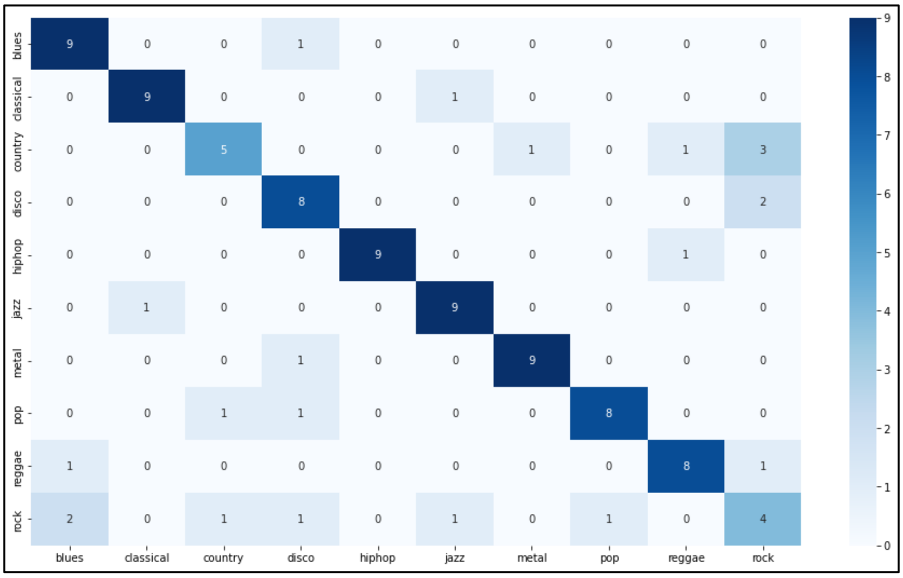


*Fig.8.11. Confusion matrix of the test dataset of GTZAN*

We got 99% of accuracy for training data, but just 78% for the testing data. We are overfitting our model. So further, we added regularization parameters and tuned other parameters to reduce this problem.

Hyperparameter tuning (or hyperparameter optimization) is the process of determining the right combination of hyperparameters that maximizes the model performance. It works by running multiple trials in a single training process. Each trial is a complete execution of a training application with values for chosen hyperparameters, set within the limits specified. This process once finished will give the set of hyperparameter values that are best suited for the model to give optimal results. In our project, we used the “hyperopt” library to help tune the parameters. After tuning the parameters, we again fitted the XGB classifier to our dataset and we were able to achieve 81% accuracy.

- ***CONVOLUTION NEURAL NETWORK:*** As machine learning and deep learning technologies maturing, the Convolutional Neural Networks (CNN) are applied to many fields, and various CNN-based variants have emerged one after another. The traditional music genre classification requires relevant professional knowledge to manually extract features from time series data. Deep learning has been proven to be effective and efficient in time series data. In order to save the user's time when searching for different styles of music, we applied CNN's advantages and characteristics in audio to implement a music genre classification model. The figures below shows the example architecture of CNN model.

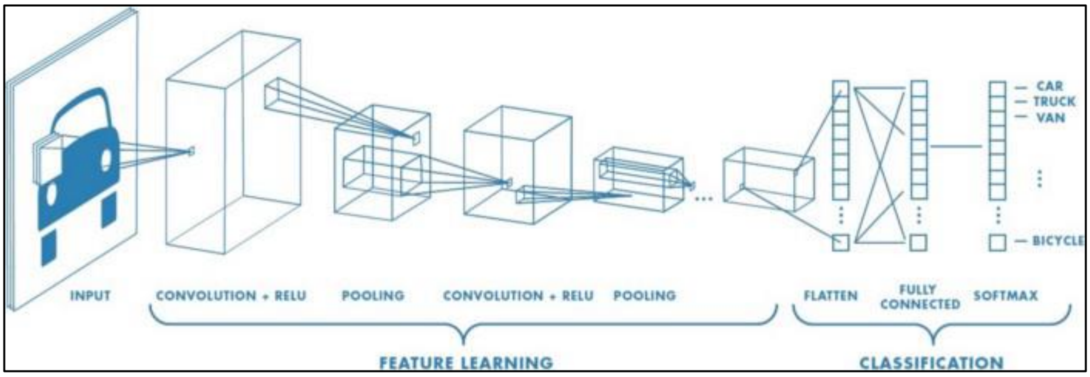


*Fig.8.12. Example1 of CNN architecture*

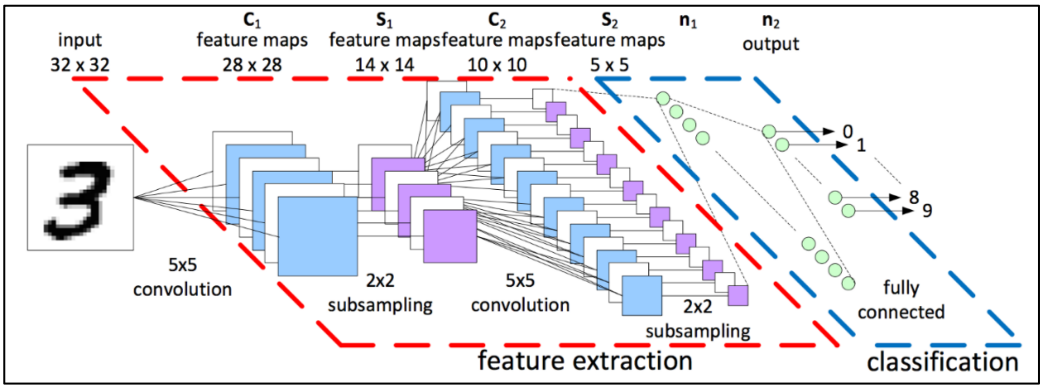


*Fig.8.13. Example2 of CNN architecture*

In a neural network, the activation function is responsible for transforming the summed weighted input from the node into the activation of the node or output for that input. The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance. In our project, we have used Relu activation factor for CNN model. It overcomes the vanishing gradient problem, allowing models to learn faster and perform better. After pre-processing the dataset and fitting the CNN model we were able to achieve 89.35% of accuracy.

# 9. RESULT

With the help of python machine learning and deep learning, XGBoost classifier gives an accuracy_score of 0.81 and Convolution Neural Network gives an accuracy_score of 0.8935.

# 10. CONCLUSION AND RECOMMENDATION

The features identified for GTZAN dataset have been very effective in discriminating the genre. All the previous efforts were aimed at using existing features instead of transformed features, which would not only reduce the number of dimensions but also be effective in discriminating the classes. Having a sample of 100 audio files per decade per genre will be more effective to improve the predictive accuracy of the models. Efforts must be made to accumulate such a collection. Most of the modern-day music is a fusion of multiple genres like blues+ classical, indie pop + metallic + jazz etc. Having additional fusion genre and related audio files will make the model more effective from a commercialization perspective. The GTZAN dataset and the genre are applicable for only western music and there are many other styles like Indian, Asian, Middle Eastern music style. Preparing a dataset with sufficient samples encompassing all the styles is a very humongous task but if accomplished will provide additional teeth to the model. After all was said and done, we attempted to perform music-genre classification using two distinct models: a XGBoost classifier and a standard convolution neural network with softmax output. With the help of the accuracy scores, we can conclude Convolution Neural Network is the best for classification of music genre GTZAN dataset among these two.

# 11. REFERENCES

1) Karunakaran N & Arya A, “A Scalable Hybrid Classifier for Music Genre Classification using Machine Learning Concepts and Spark”, International Conference on Intelligent Autonomous Systems, March 2018.

2) Deepanway Ghosal, Maheshkumar H. Kolekar, “Music Genre Recognition using Deep Neural Networks and Transfer Learning”, Indian Institute of Technology Patna, September 2018.

3) Albert Jim´enez & FerranJos´e, “Music Genre Recognition with Deep Neural Networks”, University Polit`ecnica de Catalunya, May 2017.

4) Hareesh Bahuleyan, “Music Genre Classification using Machine Learning Techniques”, University of Waterloo, ON, Canada, April 2018.

5) Vishnupriya S & Meenakshi K, “Automatic Music Genre Classification using Convolution Neural Network”, Jan 2018.

6) B. L. Sturm. “An analysis of the GTZAN music genre dataset.” In Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies, pp. 7-12. 2012.

7) B. L. Sturm. “The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use.” arXiv preprint arXiv:1306.1461. 2013.

8) A. Olteanu, “GTZAN Dataset - Music Genre Classification”, Kaggle.com, 2020. [Online]. Available: https://www.kaggle.com/andradaolteanu/gtzan-dataset-music-genreclassification.

9) GTZAN Genre Collection.

10) Dwivedi, Priya. “Using CNNs and RNNs for Music Genre Recognition.” Towards Data Science, Medium, 13 Dec. 2018, towardsdatascience.com/using-cnns-and-rnns-for-music-genre-recognition-2435fb2ed6af.

11) “Music Genre Recognition” by Karin Kosina.