MUSIC GENRE CLASSIFICATION

J COMPONENT PROJECT REPORT

Winter 2020-21

Submitted by

VIBHU KUMAR SINGH 19BCE0215

AVNISH TIWARI 19BCE0634

RINISHA JAIN 19BCE0715

RISHABH NAGAR 19BCE0722

in partial fulfilment for the award of the degree of

B. Tech

in

Computer Science and Engineering

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Vellore-632014, Tamil Nadu, India

School of Computer Science and Engineering

May, 2021

INDEX

1. Abstract:

Categorizing music files according to their genre is a challenging task in the area of music information retrieval (MIR). In this study, we compare the performance of two classes of models. The first approach utilizes hand-crafted features, both from the time domain and frequency domain and classification algorithms like SVM and KNN are used to train the classifier. The second is a deep learning approach wherein a DNN model is trained end-to-end, to predict the genre label of an audio signal, solely using its features. We will mainly focus on the Deep Learning approach and finally draw comparisons between both approaches.

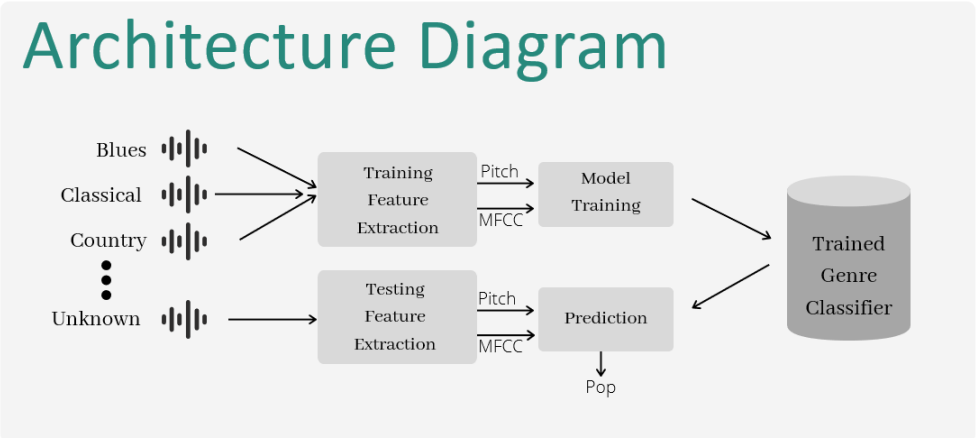
2. Introduction:

Music has become popular thanks to the growth of music streaming services. We listen to music while driving, exercising, working, or just relaxing. The role of music in eliciting emotion and processing our thoughts has not been harmed by the continuing interruption in our daily lives, as evidenced by the rise of "Zoom concerts."

The playlists, which are mostly organized by genre, are a key feature of these services. This information could come from the people who publish the songs manually naming them. However, this does not scale well and maybe abused by artists seeking to cash in on a particular genre's success. Using automatic music genre classification is a safer choice.

3. Architectural Diagram:

This is the Music Genre Classification Architectural Diagram. We divide the dataset into 3 parts train, dev and test. The Model is trained with the help of training set. Dev set act as a hyper-parameter which helps in developing the model. Test set is used to test the model. Since this model uses Deep learning, we will go for Feature extraction for training the model. We will give an unknown audio as an input and the Trained Classifier Model will help in Predicting the Genre of the Unknown Audio after its feature extraction.



4. Background Study:

4.1. Wavenet: A generative model for raw audio

We see in this paper that they introduced WaveNet, a deep neural network for generating raw audio waveforms. Despite the fact that the model is completely probabilistic and autoregressive, with each audio sample's predictive distribution conditioned on all previous ones, they show that it can be trained efficiently on data with tens of thousands of samples per second of audio.

WaveNet, a profound generative model of sound information that works straightforwardly at the waveform level. WaveNets are autoregressive and consolidate causal channels with enlarged convolutions to permit their responsive fields to develop dramatically with profundity, which is essential to display the long-range transient conditions in sound signs. They have shown how WaveNets can be adapted on different contributions to a worldwide (for example speaker personality) or neighbourhood way (for example etymological highlights).

When applied to TTS (Text to Speech), WaveNets created tests that outflank the current best TTS frameworks in emotional effortlessness. At long last, WaveNets showed exceptionally encouraging outcomes when applied to music sound displaying and discourse acknowledgment.

4.2. Imagenet classification with deep con- volutional neural networks.

In this research paper, to classify the 1.2 million high-resolution images in the ImageNet LSVRC- 2010 contest into the 1000 different groups, we trained a massive, deep convolutional neural network. We achieved top-1 and top-5 error rates of 37.5 percent and 17.0 percent, respectively, on the test results, which is significantly better than the previous state-of-the-art. Five convolutional layers, some of which are accompanied by max-pooling layers, and three fully-connected layers with a final 1000-way SoftMax make up the neural network, which has 60 million parameters and 650,000 neurons. We used non-saturating neurons and a very powerful GPU implementation of the convolution operation to make training go faster. We used a recently developed regularisation method called "dropout" to minimise overfitting in the fully-connected layers, which proved to be very successful. In the ILSVRC-2012 competition, we entered a version of this model and won with a top-5 test error rate of 15.3 percent, compared to 26.2 percent for the second-best entry.

4.3. Feature selection, L_1 vs. L_2 regularization, and rotational invariance.

We consider administered learning within the nearness of exceptionally numerous unimportant highlights and ponder two distinctive regularization strategies for

avoiding overfitting. Centring on calculated relapse, we appear that utilizing L1 regularization of the parameters, the test complexity (i.e., the number of training illustrations required to memorize "well,") develops as it were logarithmically within the number of unessential highlights. This logarithmic rate matches the leading known bounds for highlight determination, and shows that L1 regularized calculated relapse can be successful indeed in the event that there are exponentially numerous unessential highlights as there are preparing illustrations. We too grant a lower-bound appearing that any rotationally invariant algorithm---including calculated relapse with L2 regularization, SVMs, and neural systems prepared by backpropagation---has a most noticeably awful case test complexity that develops at slightest straight within the number of unimportant highlights.

4.4. Speech Recognition using MFCC

The Mel-Scale Frequency Cepstral Coefficients (MFCC) derived from the speech signal of spoken words are used in this paper to characterise a speech recognition approach. Founders prior to training and analysing speech samples using Maximum Likelihood Classifier (ML) and Support Vector Machine, Component Analysis is used as a supplement in the feature dimensional reduction state (SVM). The sixteen-ordered MFCC extracts have shown a substantial improvement in recognition rates when training the SVM with more MFCC samples randomly selected from database, compared to the ML, based on an experimental database of total 40 times of speaking words collected in an acoustically regulated room.

The theory of speech MFCC extraction for conducting word recognition was discussed in this article. The technique is outlined in detail, as well as its effectiveness. When using a help vector machine to train sentences, the training scores correlate with an increase in comprehension rates.

4.5. Power - normalized cepstral coefficients (PNCC) for robust speech recognition.

This paper describes a new feature extraction algorithm based on auditory processing called Power Normalized Cepstral Coefficients (PNCC). The use of a power-law nonlinearity, which replaces the conventional log nonlinearity used in MFCC coefficients, a noise-suppression algorithm based on asymmetric filtering to suppress background excitation, and a temporal masking module are all major new features of PNCC processing. We often recommend frequency smoothing and medium- time power analysis, in which environmental parameters are measured over a longer time period than is typically used for expression. Experiments show that PNCC processing improves speech recognition accuracy significantly as compared to MFCC and PLP processing in the presence of various types of additive noise and in reverberant environments, with only a slight increase in computational cost over conventional MFCC processing and without degrading the recognition accuracy observed

throughout training. In noisy settings, PNCC processing often outperforms techniques like the Vector Taylor Series (VTS) and the ETSI Advanced Front End (AFE) while requiring much less computation. We explain how to use "on-line proc" to implement PNCC. We define a PNCC implementation that uses "on-line processing" and does not require future input knowledge.

4.6. Support vector networks.

This research paper shows that, the value of psycho-acoustic transformations for efficient audio feature calculation is investigated in this paper. Both critical and problematic sections of the algorithm for Rhythm Patterns feature extraction are defined based on the findings. Statistical Spectrum Descriptors and Rhythm Histogram features are two new function representations introduced in this context. A music genre classification task involving three reference audio collections is used to assess both the individual and combined feature sets. On the same data sets, the results are compared to published steps. Experiments have shown that using pyscho-acoustic transitions improves classification accuracy significantly in all environments.

4.7. Chroma feature analysis and synthesis.

Since one of the most popular ways for people to manage digital music databases is by musical genre, music genre identification is a critical activity that has been studied extensively by the Music Information Retrieval (MIR) research community since 2002. We present a novel and successful approach for automated musical genre recognition based on the fusion of various sets of features in this paper. Both acoustic and visual features are taken into account, analysed, compared, and fused in a final ensemble that has classification accuracy that is equal to or better than other state-of-the- art approaches. Mel scale zoning is used to extract the visual features from sub-windows of the spectrogram: the signal received as input. Mel scale zoning extracts visual features from sub- windows of the spectrogram: the input signal is represented by its spectrogram, which is divided into sub-windows to extract local features; feature extraction is done by calculating texture descriptors and bag of features projections from each sub-window; the final decision is made using an ensemble of SVM c For the first time, we demonstrate that a bag of features approach can be successful in this problem in this paper. We propose an ensemble of heterogeneous classifiers for optimising the output that can be obtained starting from the acoustic features in terms of feature vectors. To improve recognition efficiency and reduce computational complexity, first timbre features are extracted from the audio signal, then some statistical measurements are measured from the texture window and modulation range, and finally a feature selection is performed. Finally, by combining the scores of heterogeneous classifiers, the resulting descriptors are graded (SVM and Random subspace of AdaBoost). Three well-known databases are used in the experimental evaluation: the Latin Music Database (LMD), the ISMIR 2004 database, and the

GTZAN genre list. The proposed approach's recorded output is very promising, as it outperforms other state-of-the-art approaches without requiring any ad hoc parameter optimization (i.e., using the same ensemble of classifiers and parameters setting in all the three datasets). The benefit of combining visual and audio features is also shown using Q-statistics, which show that the two sets of features are partly independent and can be fused in a heterogeneous system. The MATLAB code for the ensemble of classifiers and the extraction of visual features will be made publicly available (see footnote 1) for possible comparisons by other researchers. The code for acoustic features is not available since it is used in a commercial system.

4.8. Music type classification by spectral contrast feature.

On reviewing this research paper, it is drawn out that the management of a digital music archive benefits greatly from automatic music type classification. The use of an Octave based Spectral Contrast function to reflect the spectral characteristics of a music clip is suggested in this paper. Instead of the average spectral envelope, it reflected the relative spectral distribution. Experiments revealed that the Octave-based Spectral Contrast function was effective in classifying music types. Another comparison experiment revealed that the Octave-based Spectral Contrast function discriminates between different music types better than Mel-Frequency Cepstral Coefficients (MFCC), which was previously used in music type classification systems.

The Octave-based Spectral Contrast function was introduced in this article. Spectral Contrast reflects the relative spectral features by measuring the amplitude of spectral peaks, valleys, and differences in each sub-band.

4.9. Adam: A method for stochastic optimization.

We present Adam, a first-order gradient-based optimization algorithm for stochastic objective functions based on adaptive lower-order moment estimates. The method is simple to implement, computationally effective, requires little memory, is invariant to gradient diagonal rescaling, and is well suited for problems with large amounts of data and/or parameters. The approach can also be used to solve problems with non-stationary targets and/or very noisy and/or sparse gradients. The hyper-parameters have intuitive representations and require little tuning in most cases. There are some links to similar algorithms that Adam was influenced by. We also look at the algorithm's theoretical convergence properties and have a regret bound on the convergence rate that matches the best-known results in the online convex optimization context. Adam performs well in practise and compares favourably to other stochastic optimization approaches, according to empirical evidence. Finally, we talk about Addamax, an Adam variant based on the infinity norm.

4.10. Psychoacoustics facts and models.

Psychoacoustics – Realities and Models offers a one-of-a-kind, comprehensive outline of data depicting the handling of sound by the human hearing framework. It incorporates quantitative relations between sound jolts and sound-related recognition in terms of hearing sensations, for which quantitative models are given, as well as an unequalled collection of information on the human hearing framework as a collector of acoustic data. In expansion, numerous illustrations of the common-sense application of the comes about of fundamental inquire about in areas such as commotion control, audiology, or sound quality building are point by point. The third version incorporates an extra chapter on audio-visual intuitive and applications, furthermore more on applications all through. Audits of past versions have characterized it as "a fundamental source of psychoacoustic information," "a major point of interest," and a book that "without question will have a long-lasting impact on the standing and future advancement of this logical space."

4.11. On the Use of Zero-Crossing Rate for an Application of classification of percussive sounds:

We look at how to extract rhythm descriptors from audio signals automatically, with the aim of using them in content-based musical applications like MPEG7. Our aim is to approach auditory scene comprehension in raw polyphonic audio signals without first separating the sources. This paper suggests an approach for automatically extracting time indexes of occurrences of different percussive timbres in an audio signal as a first step toward the automated extraction of rhythmic structures from signals taken from the popular music repertoire. In conclusion, this paper discovered that the classification of percussive sounds is a specific problem within this context. This paper makes no claim to include a comprehensive analysis of instrument classification strategies. The classification task we're talking about is distinct from the identification task the reader may be familiar with (training the machine with a vast collection of marked results, followed by actual classification).

4.12. Automatic Musical Genre Classification of Audio Signals.

In this paper the authors basically explain that Musical genres are human-made labels that define various styles of music. A musical genre is described by the characteristics that its members have in common. These characteristics are usually linked to the music's instrumentation, rhythmic structure, and harmonic material. The vast collections of music available on the Internet are often organised using genre hierarchies. Automatic musical genre classification can help or even replace the human user in this process, making it a useful addition to music information retrieval systems.

Furthermore, automated musical genre classification offers a basis for designing and assessing features for any form of musical signal content-based analysis. The automated

classification of audio signals into a hierarchy of musical genres is investigated in this paper. Three feature sets are proposed for representing timbral texture, rhythmic content, and pitch content, respectively. The efficiency and relative significance of the proposed features was examined by using real-world audio collections to train statistical pattern recognition classifiers. The classification schemes are defined for both whole files and real-time frames.

The proposed paper concludes that the classification rate is 61% for ten musical genres. This finding is similar to what has been recorded for musical genre classification in humans.

4.13. Recognition of Music Types

In this paper we see they describe a music type recognition system that can be used to index and search in multimedia databases. A new approach to temporal structure modelling is supposedly known as ETM-NN (Explicit Time Modelling with Neural Network) method uses abstraction of acoustical events to the hidden units of a neural network.

They trained and analysed the models and compared mainly HMM and ETM-NN models and concluded that their approach gave them an efficiency of 86.1% Their ETM-NN approach combines discriminative power of neural networks with a direct modelling of temporal structures.

4.14. Greedy function approximation: a gradient boosting.

Automatic music classification is needed for searching and organising which digital music collections. This paper describes a new method that uses support vector machines to identify songs based on features measured over their entire lengths and was tested on the task of artist recognition. Since support vector machines are exemplar-based classifiers, it makes intuitive sense to train and classify entire songs rather than short-time features. We show that this classifier outperforms related classifiers that use only SVMs or song-level features on a dataset of 1200 pop songs performed by 18 artists. We also show that when classifiers are trained and tested on separate albums, KL divergence between single Gaussians and Mahala Nobis distance between MFCC statistics vectors work similarly, but KL divergence outperforms Mahala Nobis distance when trained and tested on the same albums.

4.15. Audio set: An ontology and human-labelled dataset for audio events.

Sound occasion acknowledgment, the human-like capacity to recognize and relate sounds from sound, is an early issue in machine insight. Tantamount issues, for example, object recognition in pictures have received huge rewards from extensive

datasets – mainly ImageNet. This paper portrays the making of Sound Set, a largescale dataset of physically explained sound occasions that attempts to overcome any issues in information accessibility among picture and sound exploration. Utilizing a painstakingly organized various levelled philosophy of 632 sound classes guided by the writing and manual curation, we gather information from human labellers to test the presence of explicit sound classes in 10 second sections of YouTube recordings. Portions are proposed for marking utilizing look through dependent on metadata, setting (e.g., connections), and substance investigation. The outcome is a dataset of remarkable expansiveness and size that will, we trust, significantly animate the advancement of superior sound occasion recognizers.

This research paper introduced the Audio Set dataset of generic audio events, comprising an ontology of 632 audio event categories and a collection of 1,789,621 labelled 10 sec excerpts from YouTube videos.

4.16. The modelling of time information for automatic genre recognition systems in audio signals

The construction of large databases arising from both the reconstruction of existing analogue records and the inclusion of new material necessitates the production of fast and increasingly accurate content analysis and description tools that can be used for searches, content queries, and interactive access.

In this paper they see Musical genres are crucial descriptors in this sense, as they have been commonly used for years to organise music catalogues, libraries, shops and other collections. Despite their widespread use, musical styles remain loosely defined concepts, rendering automatic classification a difficult task. The majority of automated genre classification models use the same pattern recognition architecture, which involves extracting features from chunks of audio signal and classifying them independently.

Instead, when classifying audio signals in terms of genre, they concentrate on low-level temporal relationships between chunks; in other words, they look for ways to model short-term time structures from background information in music segments to improve classification accuracy by reducing ambiguities. A detailed comparison of five different time modelling schemes is presented, with classification results for a database of 1400 songs evenly distributed across seven genres recorded.

We see they have compared 5 different methods taking low level, short-term time relationships into account to classify audio excerpts into musical genres and they conclude that SVMs with delayed inputs proved to give the best results with a simple modelling of time structures.

But that's not what they wanted to tell, they wanted to basically tell that a simple model somehow improves musical genre classification results in many cases. Also Reported

results can be greatly improved by considering hierarchical classification techniques to model the underlying genres.

4.17. Bagging Predictors

On reviewing this research paper, it is drawn out that the management of a digital music archive benefits greatly from automatic music type classification. The use of an Octave based Spectral Contrast function to reflect the spectral characteristics of a music clip is suggested in this paper. Instead of the average spectral envelope, it reflected the relative spectral distribution. Experiments revealed that the Octave-based Spectral Contrast function was effective in classifying music types. Another comparison experiment revealed that the Octave-based Spectral Contrast function discriminates between different music types better than Mel-Frequency Cepstral Coefficients (MFCC), which was previously used in music type classification systems. The Octave-based Spectral Contrast function was introduced in this article. Spectral Contrast reflects the relative spectral features by measuring the amplitude of spectral peaks, valleys, and differences in each sub-band.

4.18. Random Forests

This paper describes a new feature extraction algorithm based on auditory processing called Power Normalized Cepstral Coefficients (PNCC). The use of a power-law nonlinearity, which replaces the conventional log nonlinearity used in MFCC coefficients, a noise-suppression algorithm based on asymmetric filtering to suppress background excitation, and a temporal masking module are all major new features of PNCC processing. We often recommend frequency smoothing and medium- time power analysis, in which environmental parameters are measured over a longer time period than is typically used for expression. Experiments show that PNCC processing improves speech recognition accuracy significantly as compared to MFCC and PLP processing in the presence of various types of additive noise and in reverberant environments, with only a slight increase in computational cost over conventional MFCC processing and without degrading the recognition accuracy observed throughout training. In noisy settings, PNCC processing often outperforms techniques like the Vector Taylor Series (VTS) and the ETSI Advanced Front End (AFE) while requiring much less computation. We explain how to use "on-line proc" to implement PNCC. We define a PNCC implementation that uses "on-line processing" and does not require future input knowledge.

4.19. Audio Spectrogram Representations for Processing with Convolutional Neural Network.

One of the choices that emerge when planning a neural arrange for any application is how the information ought to be spoken to in arrange to be displayed to, and conceivably produced by, a neural arrange. For sound, the choice is less self-evident than it appears to be for visual pictures, and an assortment of representations has been utilized for diverse applications counting the crude digitized test stream, hand-crafted highlights, machine found highlights, MFCCs and variations that incorporate deltas, and an assortment of unearthly representations. This paper audits a few of these representations and issues that emerge, centring especially on spectrograms for producing sound utilizing neural systems for fashion exchange.

4.20. Combining visual and acoustic features for music genre classification.

Since one of the most popular ways for people to manage digital music databases is by musical genre, music genre identification is a critical activity that has been studied extensively by the Music Information Retrieval (MIR) research community since 2002. We present a novel and successful approach for automated musical genre recognition based on the fusion of various sets of features in this paper. Both acoustic and visual features are taken into account, analysed, compared, and fused in a final ensemble that has classification accuracy that is equal to or better than other state-of-the- art approaches. Mel scale zoning is used to extract the visual features from sub-windows of the spectrogram: the signal received as input. Mel scale zoning extracts visual features from sub- windows of the spectrogram: the input signal is represented by its spectrogram, which is divided into sub-windows to extract local features; feature extraction is done by calculating texture descriptors and bag of features projections from each sub-window; the final decision is made using an ensemble of SVM c for the first time, we demonstrate that a bag of features approach can be successful in this problem in this paper. We propose an ensemble of heterogeneous classifiers for optimising the output that can be obtained starting from the acoustic features in terms of feature vectors. To improve recognition efficiency and reduce computational complexity, first timbre features are extracted from the audio signal, then some statistical measurements are measured from the texture window and modulation range, and finally a feature selection is performed. Finally, by combining the scores of heterogeneous classifiers, the resulting descriptors are graded (SVM and Random subspace of AdaBoost). Three well-known databases are used in the experimental evaluation: the Latin Music Database (LMD), the ISMIR 2004 database, and the GTZAN genre list. The proposed approach's recorded output is very promising, as it outperforms other state-of-the-art approaches without requiring any ad hoc parameter optimization (i.e. using the same ensemble of classifiers and parameters setting in all the three datasets). The benefit of combining visual and audio features is also shown using Q-statistics, which show that the two sets of features are partly independent and can be fused in a heterogeneous system. The MATLAB code for the ensemble of classifiers and the extraction of visual features will be made publicly available (see

footnote 1) for possible comparisons by other researchers. The code for acoustic features is not available since it is used in a commercial system.

4.21. Song-level features and support vector machines for music classification.

Automatic music classification is needed for searching and organising which digital music collections. This paper describes a new method that uses support vector machines to identify songs based on features measured over their entire lengths and was tested on the task of artist recognition. Since support vector machines are exemplar-based classifiers, it makes intuitive sense to train and classify entire songs rather than short-time features. We show that this classifier outperforms related classifiers that use only SVMs or song-level features on a dataset of 1200 pop songs performed by 18 artists. We also show that when classifiers are trained and tested on separate albums, KL divergence between single Gaussians and Mahala Nobis distance between MFCC statistics vectors work similarly, but KL divergence outperforms Mahala Nobis distance when trained and tested on the same albums.

4.22. On the modelling of time information for automatic genre recognition systems in audio signals.

This paper proposes that the construction of massive databases resulting from both the reconstruction of existing analogue records and the addition of new material necessitates the development of fast and increasingly accurate content analysis and description tools that can be used for searches, content queries, and interactive access. Musical genres are important descriptors in this context because they have been commonly used for years to organise music catalogues, libraries, and stores. Despite their widespread use, musical styles remain loosely established concepts, making automatic classification a difficult task. The architecture of most automated genre classification models is the same: extracting features from chunks of audio signal and classifying them. The majority of automated genre classification models use the same pattern recognition architecture, which involves extracting features from chunks of audio signal and classifying them independently. Instead, when classifying audio signals in terms of genre, we concentrate on low- level temporal relationships between chunks; in other words, we look for ways to model short-term time structures from background information in music segments to improve classification accuracy by reducing ambiguities. A detailed comparison of five different time modelling schemes is presented, as well as classification results for a database of 1400 songs evenly distributed across seven genres.

4.23. Dropout: A Simple Way to Prevent Neural Networks from Overfitting.

In this paper we see the problem faced in training using deep learning and its solution. So basically, with limited training data, complicated relationships in deep learning models will be the result of sampling noise, so they will exist in the training set but not in real test data even if it is drawn from the same distribution. This leads to overfitting and can be solved with the method given known as Dropout. Dropout is a technique that addresses these issues. It prevents overfitting and provides a way of approximately combining exponentially many different neural network architectures efficiently.

4.24. Convolutional neural networks for speech recognition.

We present Adam, a first-order gradient-based optimization algorithm for stochastic objective functions based on adaptive lower-order moment estimates. The method is simple to implement, computationally effective, requires little memory, is invariant to gradient diagonal rescaling, and is well suited for problems with large amounts of data and/or parameters. The approach can also be used to solve problems with non-stationary targets and/or very noisy and/or sparse gradients. The hyper-parameters have intuitive representations and require little tuning in most cases. There are some links to similar algorithms that Adam was influenced by. We also look at the algorithm's theoretical convergence properties and have a regret bound on the convergence rate that matches the best-known results in the online convex optimization context. Adam performs well in practise and compares favourably to other stochastic optimization approaches, according to empirical evidence. Finally, we talk about Addamax, an Adam variant based on the infinity norm.

4.25. Cyclic Tempogram – A Mid-level tempo representation for Music Signals

In accordance to this paper, extraction of local tempo and beat information from audio recordings is a difficult job, particularly when the music has a lot of tempo variations. Furthermore, the presence of different pulse levels including measure, tactus, and tatum makes determining absolute tempo difficult. We present a robust mid-level representation for encoding local tempo information in this paper. We introduce the concept of cyclic tempograms, which are tempi that vary by a power of two, similar to the well-known concept of cyclic chroma features, where pitches varying by octaves are defined.

The cyclic tempograms are the tempo-based equivalents of the harmony-based chromagrams, as previously mentioned. The cyclic models of tempograms are more resistant to tempo ambiguities created by different pulse speeds than traditional tempograms. Furthermore, cyclically changing a cyclic tempogram may be used to simulate tempo shifts.

4.26. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences.

The Mel-Scale Frequency Cepstral Coefficients (MFCC) derived from the speech signal of spoken words are used in this paper to characterise a speech recognition approach. Founders prior to training and analysing speech samples using Maximum Likelihood Classifier (ML) and Support Vector Machine, Component Analysis is used as a supplement in the feature dimensional reduction state (SVM). The sixteen-ordered MFCC extracts have shown a substantial improvement in recognition rates when training the SVM with more MFCC samples randomly selected from database, compared to the ML, based on an experimental database of total 40 times of speaking words collected in an acoustically regulated room. The theory of speech MFCC extraction for conducting word recognition was discussed in this article. The technique is outlined in detail, as well as its effectiveness. When using a help vector machine to train sentences, the training scores correlate with an increase in comprehension rates.

4.27. Converting video formats with ffmpeg.

Today's reasonable advanced video cameras have put the control of computerized recording inside most people's reach. Shockingly, this has been going with a comparing increment within the assortment of record designs and codecs accessible. A few of these groups are more productive than others, and a few are less burdened by exclusive permitting limitations. So, having the capacity to change over from one format to another may be a extraordinary offer assistance, as you'll be able choose what organize you're comfortable with and utilize that one rather than being confined to a particular record format. FFmpeg could be a simple and clear application that permits Linux clients to change over video records effectively between a assortment of distinctive groups. In this article, I walk you through introducing FFmpeg and give some teacher illustrations to illustrate the run of applications for which it can be utilized.

4.28. Parallel Convolutional Neural Networks for Music Genre and Mood Classification

Our method for detecting genre, mood, and composer in the MIREX 2016 Train/Test Classification Tasks is based on a combination of Mel spectrogram converted audio and Convolutional Neural Networks (CNN). We use two alternative CNN designs, a sequential and a parallel one, with the latter recording both temporal and timbral information in two separate streams that are later merged. After a series of trials, the critical CNN parameters such as filter kernel widths and pooling sizes were carefully set in both scenarios.

4.29. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification

This research paper shows that, the value of psycho-acoustic transformations for efficient audio feature calculation is investigated in this paper. Both critical and problematic sections of the algorithm for Rhythm Patterns feature extraction are defined based on the findings. Statistical Spectrum Descriptors and Rhythm Histogram features are two new function representations introduced in this context. A music genre classification task involving three reference audio collections is used to assess both the individual and combined feature sets. On the same data sets, the results are compared to published steps. Experiments have shown that using psycho-acoustic transitions improves classification accuracy significantly in all environments.

4.30. Automatic Musical Pattern Feature Extraction Using Convolutional Neural Network

Music class classification has been a challenging however promising assignment within the field of music information recovery (MIR). Due to the profoundly tricky characteristics of sound melodic information, retrieving informative and dependable highlights from sound signals is crucial to the execution of any music class classification framework. Past work on sound music genre classification frameworks primarily concentrated on using timbral highlights, which limits the execution. To address this issue, we propose a novel approach to extract melodic design highlights in sound music using convolutional neural organize (CNN), a show widely adopted in picture data recovery assignments. Our experiments appear that CNN has solid capacity to capture instructive highlights from the varieties of musical designs with negligible earlier information provided.

4.31. Shape quantization and recognition with randomized trees. Neural computation

In this research paper, to classify the 1.2 million high-resolution images in the ImageNet LSVRC- 2010 contest into the 1000 different groups, we trained a massive, deep convolutional neural network. We achieved top-1 and top-5 error rates of 37.5 percent and 17.0 percent, respectively, on the test results, which is significantly better than the previous state-of-the-art. Five convolutional layers, some of which are accompanied by max-pooling layers, and three fully-connected layers with a final 1000-way SoftMax make up the neural network, which has 60 million parameters and 650,000 neurons. We used non-saturating neurons and a very powerful GPU implementation of the convolution operation to make training go faster. We used a recently developed regularisation method called "dropout" to minimise overfitting in the fully-connected layers, which proved to be very successful. In the ILSVRC-2012 competition, we entered a version of this model and won with a top-5 test error rate of 15.3 percent, compared to 26.2 percent for the second-best entry.

5. Methodology:

Just by looking at the Architecture Diagram, one can understand the basic functioning of Genre Prediction using audio features. Hereby, we are proving a detailed methodology of our project:

5.1. Dataset Generation

The dataset can be generated by any of the following methods:

a) By using a dataset of audio files and extracting all audio features manually using Librosa library in python.

b) Or use an already generated csv file containing the extracted features as columns.

Kaggle has a generated dataset for Music Genre Classification which contains Audio features from about 1,000 audio files, 100 for each genre. To further increase accuracy, each audio file, which was originally 30 seconds long was split in lengths of 3 seconds, making the dataset 10 times bigger.

5.2. Training

The model is trained using two deep learning models with validation accuracy of 87 and 91% respectively.

5.3. User input Testing

The idea is that the user uploads an audio file and its audio features are extracted automatically using the function getmetadata() from Metadata.py. This function returns an array of features and this array is inputted into the trained model (model 2) after normalisation. The model outputs the name of the Genre using .predict function.

6. Proposed Model:

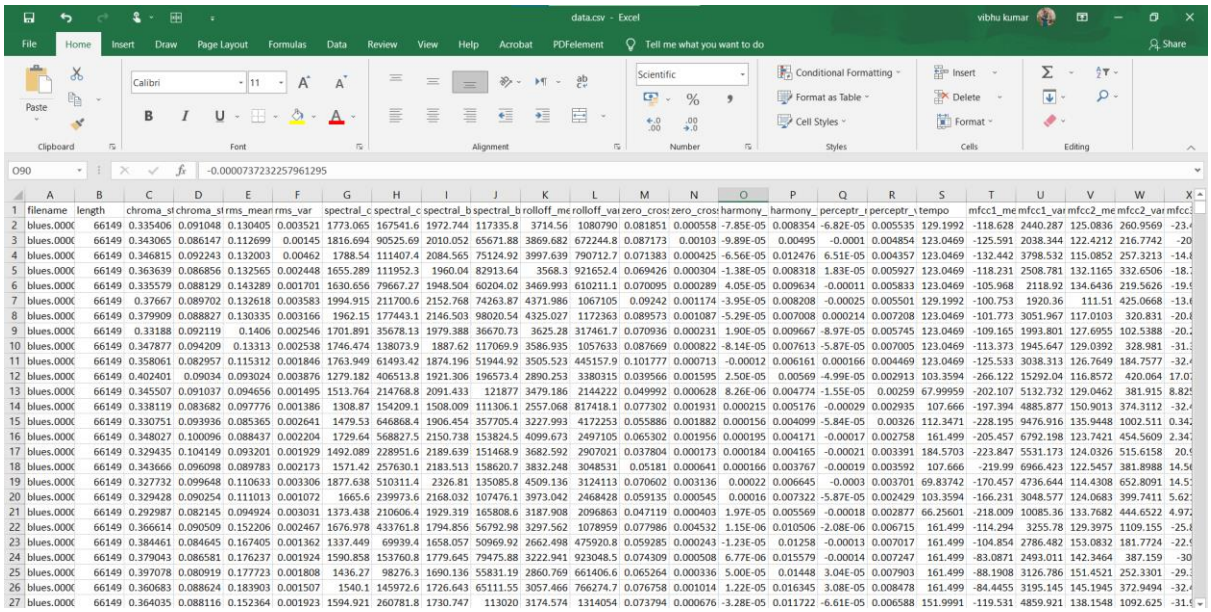
As we have mentioned before, we are going for a deep learning approach as follows:

6.1. Module 1: Defining the dataset.

To begin, we must download the libraries needed for music genre classification, such as NumPy, pandas, matplotlib, seaborn, librosa, and all other libraries.

Librosa is a Python package that analyses music and audio. It contains the components needed to construct music information retrieval systems.

A data set (or dataset) is a collection of data. In the case of tabular data. For reading the csv file we used Pandas library. The dataset has **9990** rows and **60** columns. It contains 10 genres and each genre has approximately 1000 tracks.



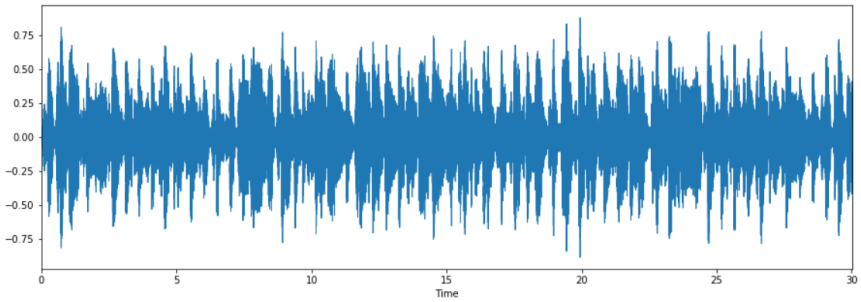
filename	length	chroma_st	chroma_st	rms_mean	rms_var	spectral_c	spectral_c	spectral_b	spectral_b	rolloff_me	rolloff_var	zero_cross	zero_cross	harmony_	harmony_	perceptr_	perceptr_	tempo	mfcc1_me	mfcc1_var	mfcc2_me	mfcc2_var	mfcc
blues.000(	66149	0.335406	0.091048	0.130405	0.003521	1773.065	167541.6	1972.744	117335.8	3714.56	1080790	0.081851	0.000558	-7.85E-05	0.008354	-6.82E-05	0.005535	129.1992	-118.628	2440.287	125.0836	260.9569	-23.4
blues.000(	66149	0.343065	0.086147	0.112699	0.00145	1816.694	90525.69	2010.052	65671.88	3869.682	672244.8	0.087173	0.00103	-9.89E-05	0.00495	-0.0001	0.004854	123.0469	-125.591	2038.344	122.4212	216.7742	-20
blues.000(	66149	0.346815	0.092243	0.132003	0.00462	1788.54	111407.4	2084.565	75124.92	3997.639	790712.7	0.071383	0.000425	-6.56E-05	0.012476	6.51E-05	0.004357	123.0469	-132.442	3798.532	115.0852	257.3213	-14.8
blues.000(	66149	0.363639	0.086856	0.132565	0.002448	1655.289	111952.3	1960.04	82913.64	3568.3	921652.4	0.069426	0.000304	-1.38E-05	0.008318	1.83E-05	0.005927	123.0469	-118.231	2508.781	132.1165	332.6506	-18.1
blues.000(	66149	0.335579	0.088129	0.143289	0.001701	1630.656	79667.27	1948.504	60204.02	3469.993	610211.1	0.070095	0.000289	4.05E-05	0.009634	-0.00011	0.005833	123.0469	-105.968	2118.92	134.6436	219.5626	-19.5
blues.000(	66149	0.37667	0.089702	0.132618	0.003583	1994.915	211700.6	2152.768	74263.87	4371.986	1067105	0.09242	0.001174	-3.95E-05	0.008208	-0.00025	0.005501	129.1992	-100.753	1920.36	111.51	425.0668	-13.6
blues.000(	66149	0.379909	0.088827	0.130335	0.003166	1962.15	177443.1	2146.503	98020.54	4325.027	1172363	0.089573	0.001087	-5.29E-05	0.007008	0.000214	0.007208	123.0469	-101.773	3051.967	117.0103	320.831	-20.8
blues.000(	66149	0.33188	0.092119	0.1406	0.002546	1701.891	35678.13	1979.388	36670.73	3625.28	317461.7	0.070936	0.000231	1.90E-05	0.009667	-8.97E-05	0.005745	123.0469	-109.165	1993.801	127.6955	102.5388	-20.2
blues.000(	66149	0.347877	0.094209	0.13313	0.002538	1746.474	138073.9	1887.62	117069.9	3586.935	1057633	0.087669	0.000822	-8.14E-05	0.007613	-5.87E-05	0.007005	123.0469	-113.373	1945.647	129.0392	328.981	-31.3
blues.000(	66149	0.358061	0.082957	0.115312	0.001846	1763.949	61493.42	1874.196	51944.92	3505.523	445157.9	0.101777	0.000713	-0.00012	0.006161	0.000166	0.004469	123.0469	-125.533	3038.313	126.7649	184.7577	-32.4
blues.000(	66149	0.402401	0.09034	0.093024	0.003876	1279.182	406513.8	1921.306	196573.4	2890.253	3380315	0.039566	0.001595	2.50E-05	0.00569	-4.99E-05	0.002913	103.3594	-266.122	15292.04	116.8572	420.064	17.0
blues.000(	66149	0.345507	0.091037	0.094656	0.001495	1513.764	214768.8	2091.433	121877	3479.186	2144222	0.049992	0.000628	8.26E-06	0.004774	-1.55E-05	0.00259	67.99959	-202.107	5132.732	129.0462	381.915	8.825
blues.000(	66149	0.338119	0.083682	0.097776	0.001386	1308.87	154209.1	1508.009	111306.1	2557.068	817418.1	0.077302	0.001931	0.000215	0.005176	-0.00029	0.002935	107.666	-197.394	4885.877	150.9013	374.3112	-32.4
blues.000(	66149	0.330751	0.093936	0.085365	0.002641	1479.53	646868.4	1906.454	357705.4	3227.993	4172253	0.055886	0.001882	0.000156	0.004099	-5.84E-05	0.00326	112.3471	-228.195	9476.916	135.9448	1002.511	0.342
blues.000(	66149	0.348027	0.100096	0.088437	0.002204	1729.64	568827.5	2150.738	153824.5	4099.673	2497105	0.065302	0.001956	0.000195	0.004171	-0.00017	0.002758	161.499	-205.457	6792.198	123.7421	454.5609	2.341
blues.000(	66149	0.329435	0.104149	0.093201	0.001929	1492.089	228951.6	2189.639	151468.9	3682.592	2907021	0.037804	0.000173	0.000184	0.004165	-0.00021	0.003391	184.5703	-223.847	5531.173	124.0326	515.6158	20.9
blues.000(	66149	0.343666	0.096098	0.089783	0.002173	1571.42	257630.1	2183.513	158620.7	3832.248	3048531	0.05181	0.000641	0.000166	0.003767	-0.00019	0.003592	107.666	-219.99	6966.423	122.5457	381.8988	14.56
blues.000(	66149	0.327732	0.099648	0.110633	0.003306	1877.638	510311.4	2326.81	135085.8	4509.136	3124113	0.070602	0.003136	0.00022	0.006645	-0.0003	0.003701	69.83742	-170.457	4736.644	114.4308	652.8091	14.51
blues.000(	66149	0.329428	0.090254	0.111013	0.001072	1665.6	239973.6	2168.032	107476.1	3973.042	2468428	0.059135	0.000545	0.00016	0.007322	-5.87E-05	0.002429	103.3594	-166.231	3048.577	124.0683	399.7411	5.621
blues.000(	66149	0.292987	0.082145	0.094924	0.003031	1373.438	210606.4	1929.319	165808.6	3187.908	2096863	0.047119	0.000403	1.97E-05	0.005569	-0.00018	0.002877	66.25601	-218.009	10085.36	133.7682	444.6522	4.971
blues.000(	66149	0.366614	0.090509	0.152206	0.002467	1676.978	433761.8	1794.856	56792.98	3297.562	1078959	0.077986	0.004532	1.15E-06	0.010506	-2.08E-06	0.006715	161.499	-114.294	3255.78	129.3975	1109.155	-25.8
blues.000(	66149	0.384461	0.084645	0.167405	0.001362	1337.449	69939.4	1658.057	50969.92	2662.498	475920.8	0.059285	0.000243	-1.23E-05	0.01258	-0.00013	0.007017	161.499	-104.854	2786.482	153.0832	181.7724	-22.9
blues.000(	66149	0.379043	0.086581	0.176237	0.001924	1590.858	153760.8	1779.645	79475.88	3222.941	923048.5	0.074309	0.000508	6.77E-06	0.015579	-0.00014	0.007247	161.499	-83.0871	2493.011	142.3464	387.159	-30
blues.000(	66149	0.397078	0.080919	0.177723	0.001808	1436.27	98276.3	1690.136	55831.19	2860.769	661406.6	0.065264	0.000336	5.00E-05	0.01448	3.04E-05	0.007903	161.499	-88.1908	3126.786	151.4521	252.3301	-29.3
blues.000(	66149	0.360683	0.088624	0.183903	0.001507	1540.1	145972.6	1726.643	65111.55	3057.466	766274.7	0.076758	0.001014	1.22E-05	0.016345	3.08E-05	0.008478	161.499	-84.4455	3195.145	145.1945	372.9494	-32.4
blues.000(	66149	0.364035	0.088116	0.152364	0.001923	1594.921	260781.8	1730.747	113020	3174.574	1314054	0.073794	0.000676	-3.28E-05	0.011722	-6.61E-05	0.006588	151.9991	-119.531	4859.921	138.1548	1092.625	-31.9

(this dataset contains 9990 rows and 60 cloumns)

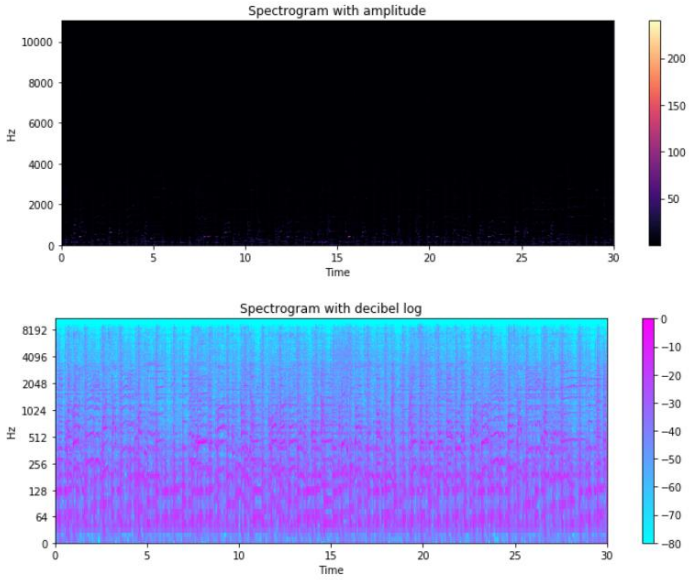
6.2. Module 2: Music Feature Classification:

To describe and put some emphasis on the features of the audio files, we have plotted a few graphs like:

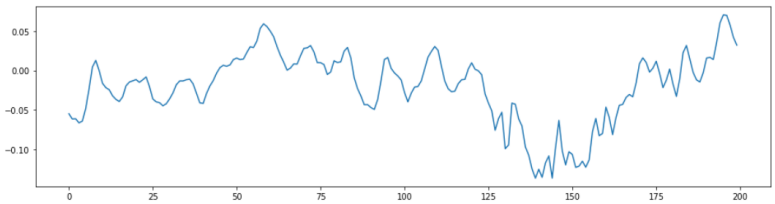
a) Wave plot: We will first load the audio file using librosa and will collect the data array and sampling rate for the audio file. Sound is a continuous wave. We can digitise sound by breaking the continuous wave into discrete signals.



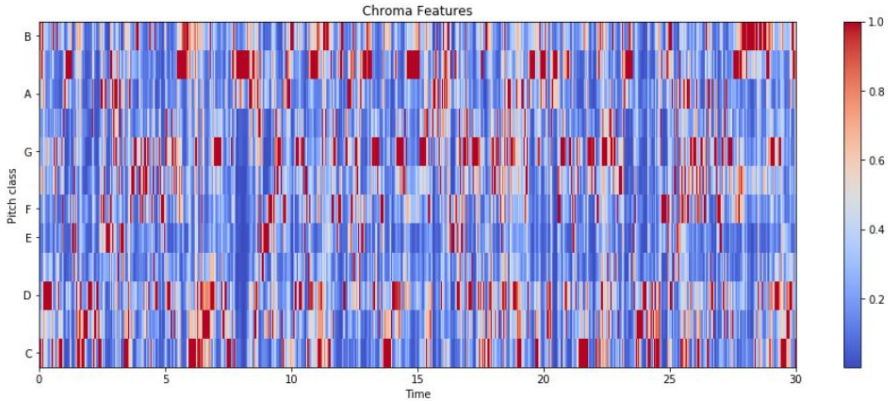
b) Spectrograms: A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time. When applied to an audio signal, spectrograms are sometimes called sonographs, voiceprints, or voicegrams. We have different spectrograms for different attributes. one axis represents the time, the second axis represents frequencies and the colours represent magnitude (amplitude) of the observed frequency at a particular time. The warm colours tend to have higher intensities and the cold colours represent lower intensities.



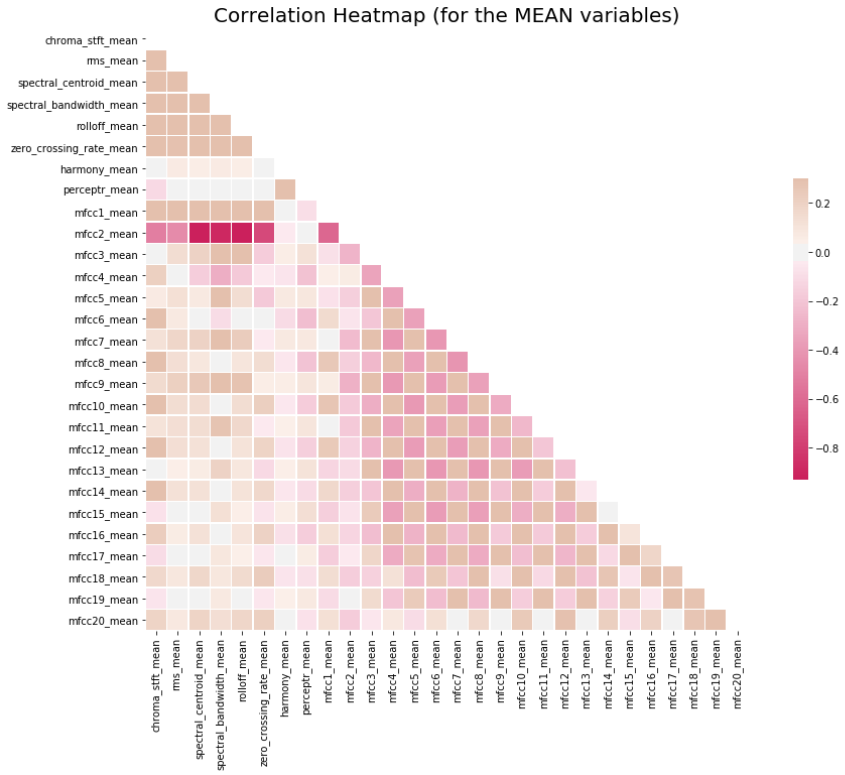
c) Audio wave: The sound source creates vibrations in the surrounding medium. As the source continues to vibrate the medium, the vibrations propagate away from the source at the speed of audio, thus forming the audio wave. One axis represents the amplitude while the other represents the time duration.



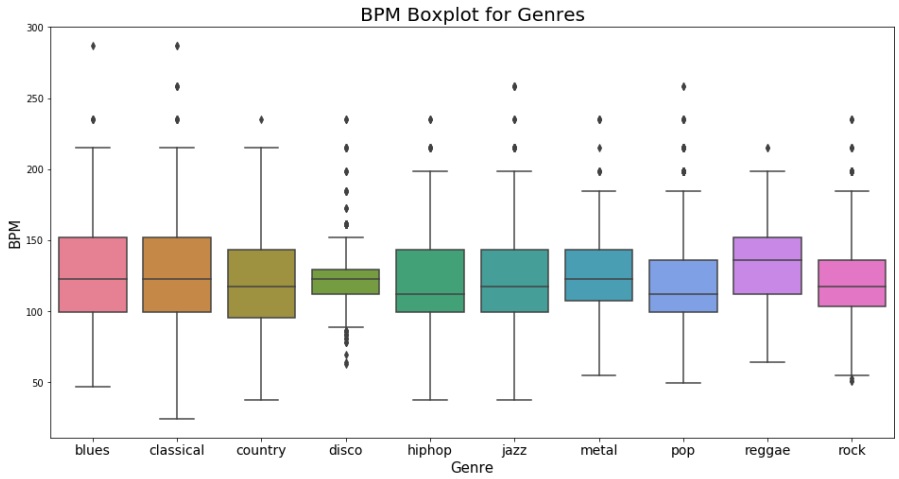
d) Chroma features: Chroma features are an interesting and powerful representation for music audio in which the entire spectrum is projected onto 12 bins representing the 12 distinct semitones (or chroma) of the musical octave. Since, in music, notes exactly one octave apart are perceived as particularly similar, knowing the distribution of chroma even without the absolute frequency (i.e. the original octave) can give useful musical information about the audio. It displays pitch vs time graph and the scale represents the intensity, warmer the colour higher the intensity.



e) Correlation Heatmap: A correlation heatmap is a heatmap that shows a 2D correlation matrix between two discrete dimensions, using coloured cells to represent data from usually a monochromatic scale. The values of the first dimension appear as the rows of the table while of the second dimension as a column. Higher the positive correlation heatmap, more will be the dependency and stronger will be the model.



f) Boxplot: **Boxplot** is a method for graphically depicting groups of numerical data through their quartiles. Box plots may also have lines extending from the boxes (*whiskers*) indicating variability outside the upper and lower quartiles. Outliers may be plotted as individual points. This is graph which is plotted as Beats per minute vs Genre.

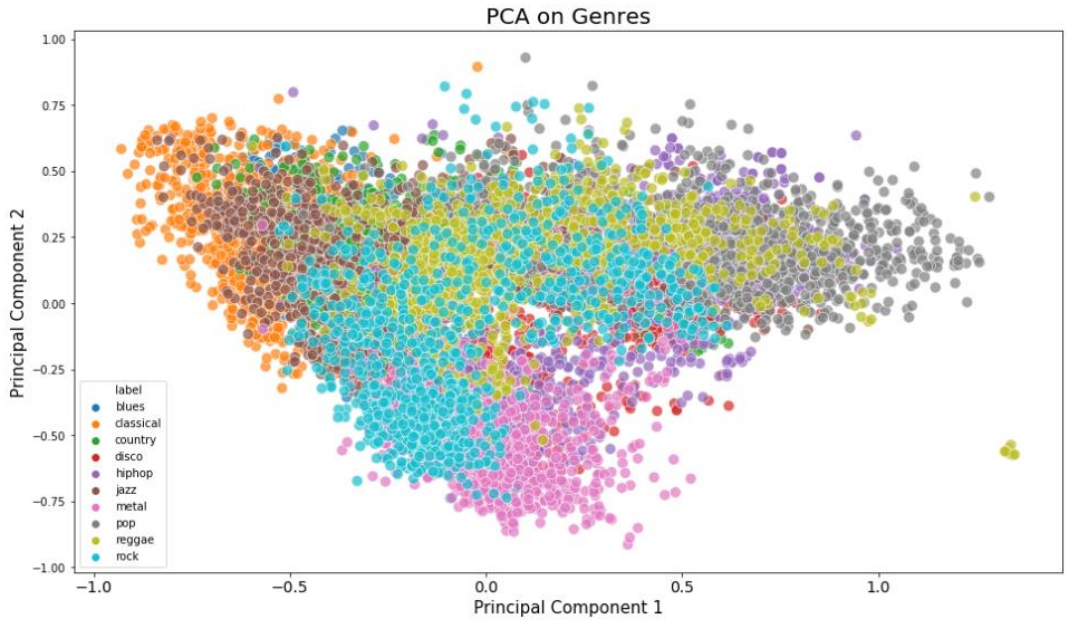


6.3. Module 3: Normalisation and Splitting the Dataset:

This module describes how the dataset is normalised which further helps in decreasing the model training runtime by a huge factor. After that the labels are divided into two PCA component to plot a scatter plot between them. That Scatterplot will consist of different colours representing the genres in the dataset.

We will make dictionary with key from 0-9 and value will consist of genre.

Now the dataset is divided into 3 parts Train, Dev, and Test with 70%, 20%, and 10% part of the dataset respectively. Now with the help of StandardScalar() we will standardizes a feature by subtracting the mean and then scaling to unit variance and will also assign the data frames to the train, dev and test data set.



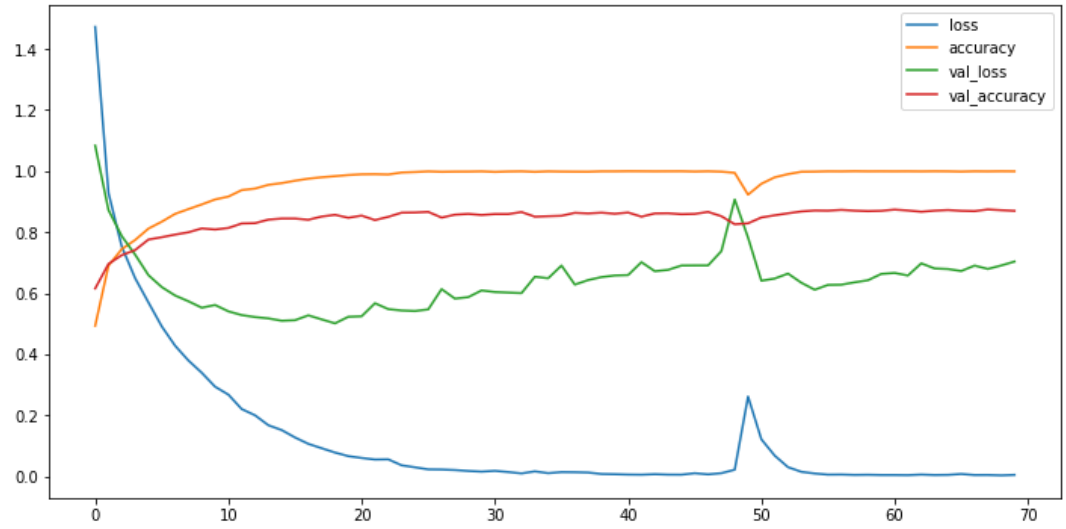
6.4. Module 4: Deep learning models training:

We then process this dataset and prepare the dataset for training purpose. The dataset is normalised using Min-Max-Scaler() and then split into train, dev and test.

The dataset was trained on the following to models:

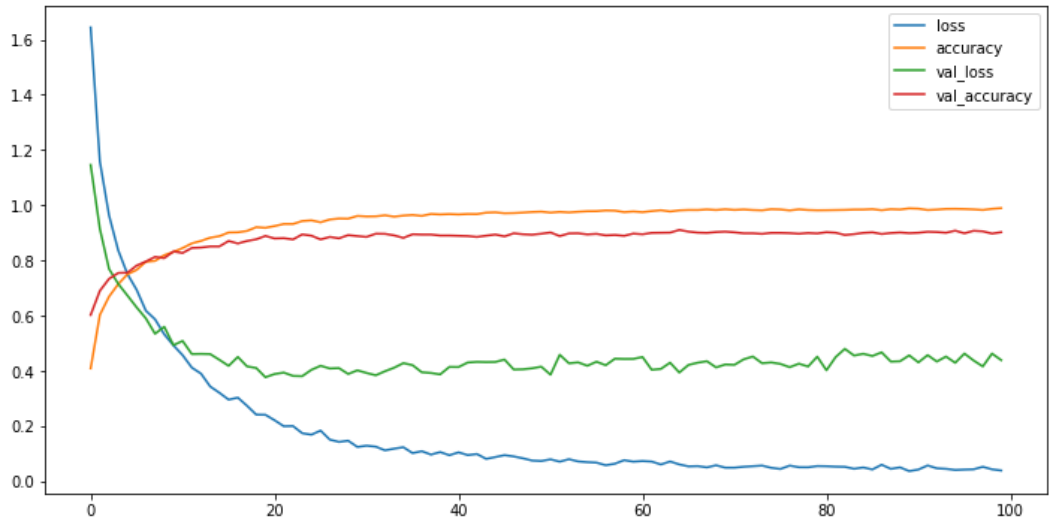
a) In the first deep learning model, we decided to make 4 dense neural layers, with 256, 128, 64 and 10 units respectively. The first three layers were activated using the 'relu' algorithm which basically return 0, if the output of the unit comes out as negative. In the last layer, there are 10 units with each unit depicting each genre of music to be classified and the activation for this layer is 'softmax' as it returns the probability of each unit. The model was run for 70 epochs.
As a result, we were able to achieve a maximum validation accuracy of 87.47%.

```
Max. Validation Accuracy 0.8741152882575989
```



b) In the second model however, we decided to go for an extra-layer with 512 units. Furthermore, we increased the dropout for each layer to 0.2, which essentially means that only 4 out of 5 units will get fired while training. This was done to avoid overfitting. The model was run for 100 epochs.
As a result, we were able to achieve a maximum validation accuracy of 91.05%, almost 4% more than the previous model.

```
Max. Validation Accuracy 0.9105156660079956
```



6.5. Module 5: Testing:

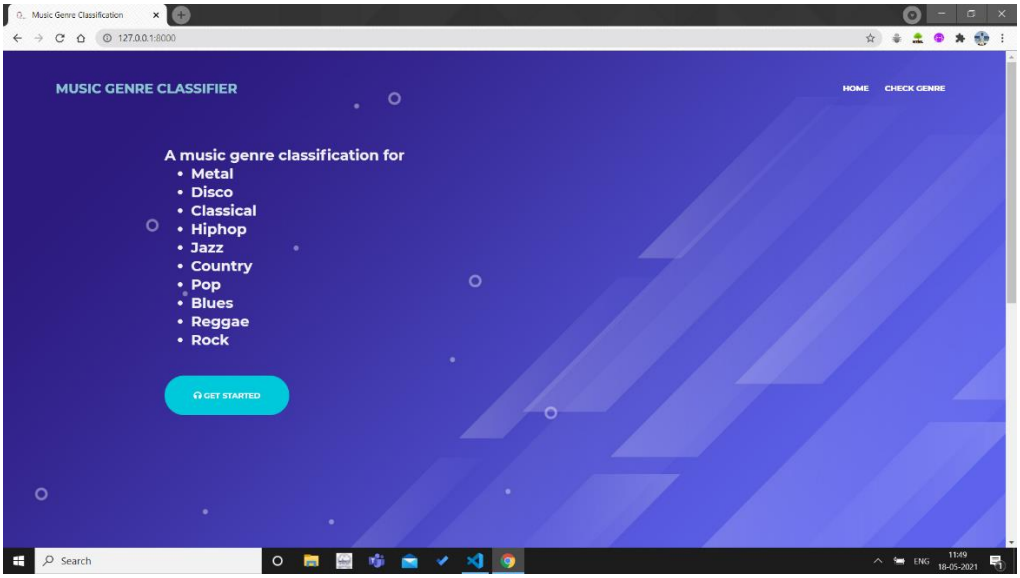
This Module will help us in predicting the genre of the Audio file. We will call library getmetadata() which will extract the required input features of the unknown file and will directly provide it to our model which will further help in predicting the genre of the Audio file.

The output provided by the model will be the number between 0-9 which will help us in calling the dictionary and then helps in recognizing the value assigned for that number. And Hence the Genre is predicted.

```
In [33]: d1 = np.array(a)
         data1 = pd.DataFrame(scaler.fit_transform([d1]))
         genre_prediction = model_2.predict(data1)
         ans = genre_prediction[0]
         ans_max=max(ans)
         index=0
         for i in range(9):
             if ans_max==ans[i]:
                 index=i
         print(index_label[index])

         disco
```

6.6. Module 6: Implementation:

In this final step, to show the working of all these modules, we developed a website where we can put any audio file in .wav format and the website can predict its' genre.



7. Comparison with existing method:

We discovered techniques like K-Nearest Neighbours (KNN) and Support Vector Machines (SVM), which are basic Machine Learning algorithms like Logistic Regression/Linear regression. So we tried these algorithms on our dataset, but we only got a validation accuracy of 65 to 73 percent, which is very poor.

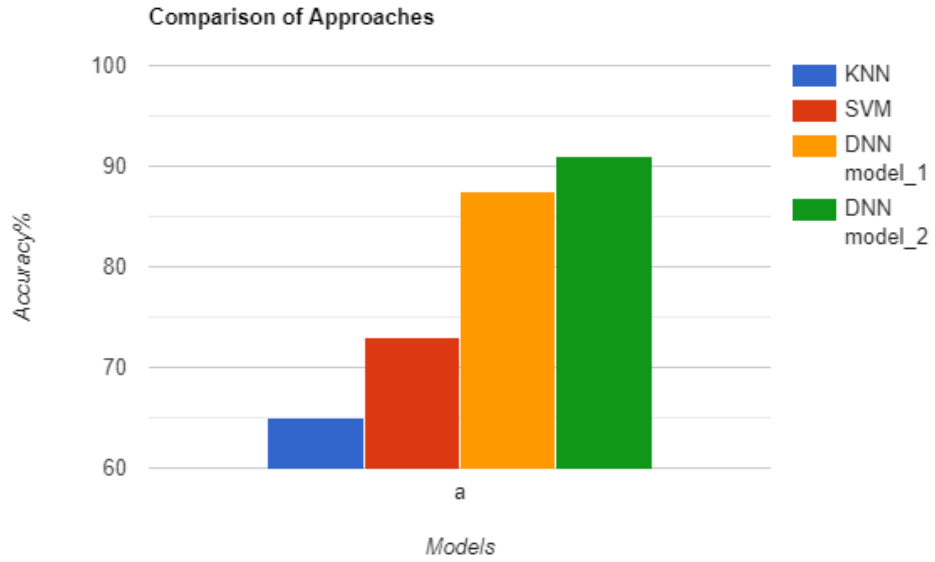
Therefore, we chose to train our model using Deep Learning instead, and we were able to achieve 91 percent validation accuracy.

As a result, we can conclude that the Deep Learning method is more effective than the current Machine Learning method. Our project's distinguishing attribute is this.

8. Result and Discussion:

8.1. Detailed explanation about the result:

As a result, we were able to build a model which predicts the genre of an audio file with an accuracy of 91%.



With the help of this graph, we can visualise which approach is better in terms of validation accuracy. It does not need reiteration that the Deep Learning approach has proved to be far better than the existing approaches in Music Genre Prediction.

Future Scope of our project : This project is root for building an even more complicated Music Recommendation System. If the validation accuracy is somehow further bumped up to 94-97%, this approach has a potential to be deployed in Music Recommendation Systems such as Youtube Music, Saavn, Amazon Music, Spotify, etc.

8.2. Sample Source Code:

Sample Code from iPyNB:

```
df = pd.read_csv('data.csv')
df.head()
data = df.iloc[0:, 1:]
y = data['label']
X = data.loc[:, data.columns != 'label']

# normalize
cols = X.columns
min_max_scaler = skp.MinMaxScaler()
np_scaled = min_max_scaler.fit_transform(X)
X = pd.DataFrame(np_scaled, columns = cols)

# Top 2 pca components
from sklearn.decomposition import PCA
```

```
pca = PCA(n_components=2)
principalComponents = pca.fit_transform(X)
principalDf = pd.DataFrame(data = principalComponents, columns = ['pc1',
'pc2'])

# concatenate with target label
finalDf = pd.concat([principalDf, y], axis = 1)

plt.figure(figsize = (16, 9))
sns.scatterplot(x = "pc1", y = "pc2", data = finalDf, hue = "label", alpha
= 0.7, s = 100);

plt.title('PCA on Genres', fontsize = 20)
plt.xticks(fontsize = 14)
plt.yticks(fontsize = 10);
plt.xlabel("Principal Component 1", fontsize = 15)
plt.ylabel("Principal Component 2", fontsize = 15)
plt.savefig("PCA_Scattert.png")

# map labels to index
label_index = dict()
index_label = dict()
for i, x in enumerate(df.label.unique()):
    label_index[x] = i
    index_label[i] = x
print(label_index)
print(index_label)

ACCURACY_THRESHOLD = 0.94

class myCallback(k.callbacks.Callback):
    def on_epoch_end(self, epoch, logs={}):
        if(logs.get('val_accuracy') > ACCURACY_THRESHOLD):
            print("\n\nStopping training as we have reached %2.2f%%
accuracy!" %(ACCURACY_THRESHOLD*100))
            self.model.stop_training = True

def trainModel(model, epochs, optimizer):
    batch_size = 128
    callback = myCallback()
    model.compile(optimizer=optimizer,
                  loss='sparse_categorical_crossentropy',
                  metrics='accuracy'
    )
    return model.fit(X_train, y_train, validation_data=(X_dev, y_dev),
epochs=epochs,
                     batch_size=batch_size, callbacks=[callback])

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

model_1 = k.models.Sequential([
```

```
    k.layers.Dense(256, activation='relu',
input_shape=(X_train.shape[1],)),
    k.layers.Dense(128, activation='relu'),
    k.layers.Dense(64, activation='relu'),
    k.layers.Dense(10, activation='softmax'),
])
print(model_1.summary())
model_1_history = trainModel(model=model_1, epochs=70, optimizer='adam')

model_2 = k.models.Sequential([
    k.layers.Dense(512, activation='relu',
input_shape=(X_train.shape[1],)),
    k.layers.Dropout(0.2),

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

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

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

    k.layers.Dense(10, activation='softmax'),
])
print(model_2.summary())
model_2_history = trainModel(model=model_2, epochs=100, optimizer='adam')
```

Sample Code from Metadata.py:

```
def getmetadata(filename):
    import librosa
    import numpy as np

    y, sr = librosa.load(filename)

    #chroma_stft

    chroma_stft = librosa.feature.chroma_stft(y=y, sr=sr)

    #rmse

    rmse = librosa.feature.rms(y=y)

    #fetching spectral centroid

    spec_centroid = librosa.feature.spectral_centroid(y, sr=sr)[0]

    #spectral bandwidth

    spec_bw = librosa.feature.spectral_bandwidth(y=y, sr=sr)

    #fetching spectral rolloff
```

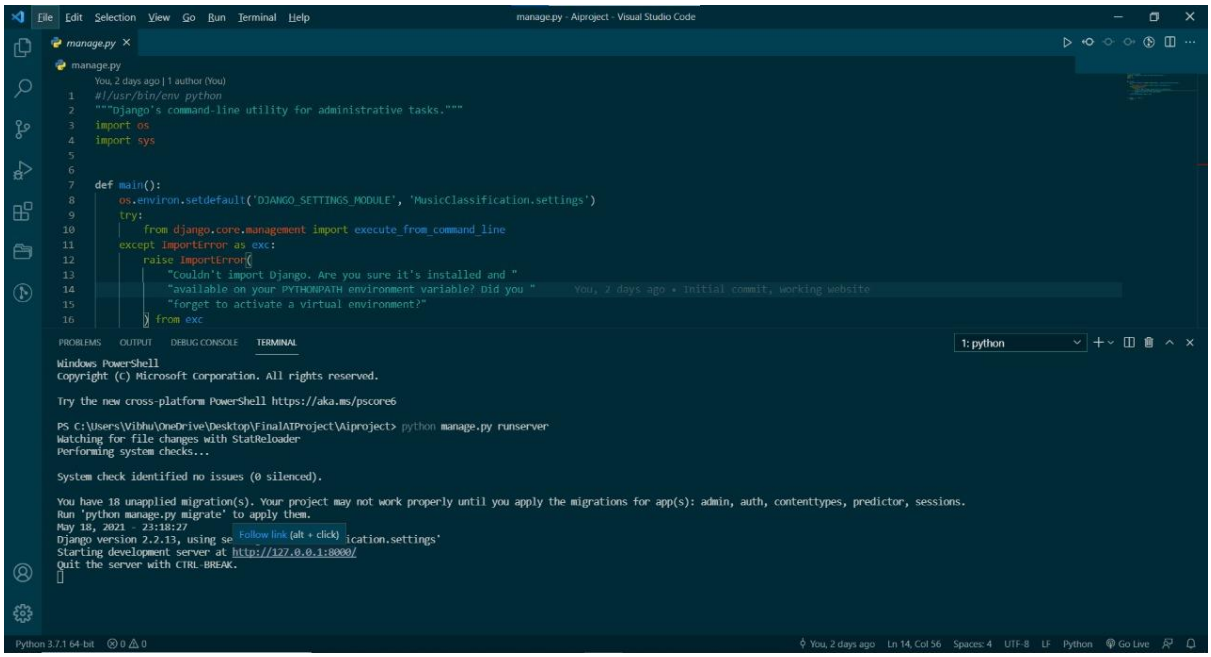
```
    spec_rolloff = librosa.feature.spectral_rolloff(y+0.01, sr=sr)[0]

    #zero crossing rate

    zero_crossing = librosa.feature.zero_crossing_rate(y)

    #fetching tempo

    onset_env = librosa.onset.onset_strength(y, sr)
    tempo = librosa.beat.tempo(onset_envelope=onset_env, sr=sr)

    #mfcc

    mfcc = librosa.feature.mfcc(y=y, sr=sr)

    #metadata dictionary

    metadata_dict =
{'chroma_stft_mean':np.mean(chroma_stft),'chroma_stft_var':np.var(chroma_s
tft),
                      'rms_mean':np.mean(rmse),'rms_var':np.var(rmse),

'spectral_centroid_mean':np.mean(spec_centroid),'spectral_centroid_var':np
.var(spec_centroid),

'spectral_bandwidth_mean':np.mean(spec_bw),'spectral_bandwidth_var':np.var
(spec_bw),

'rolloff_mean':np.mean(spec_rolloff),'rolloff_var':np.var(spec_rolloff),

'zero_crossing_rate_mean':np.mean(zero_crossing),'zero_crossing_rate_var':
np.var(zero_crossing),
                      'tempo':np.mean(tempo)}

    for i in range(1,21):
        metadata_dict.update({'mfcc'+str(i)+'_mean':np.mean(mfcc[i-
1]),'mfcc'+str(i)+'+var':np.var(mfcc[i-1])})

    return list(metadata_dict.values())
```

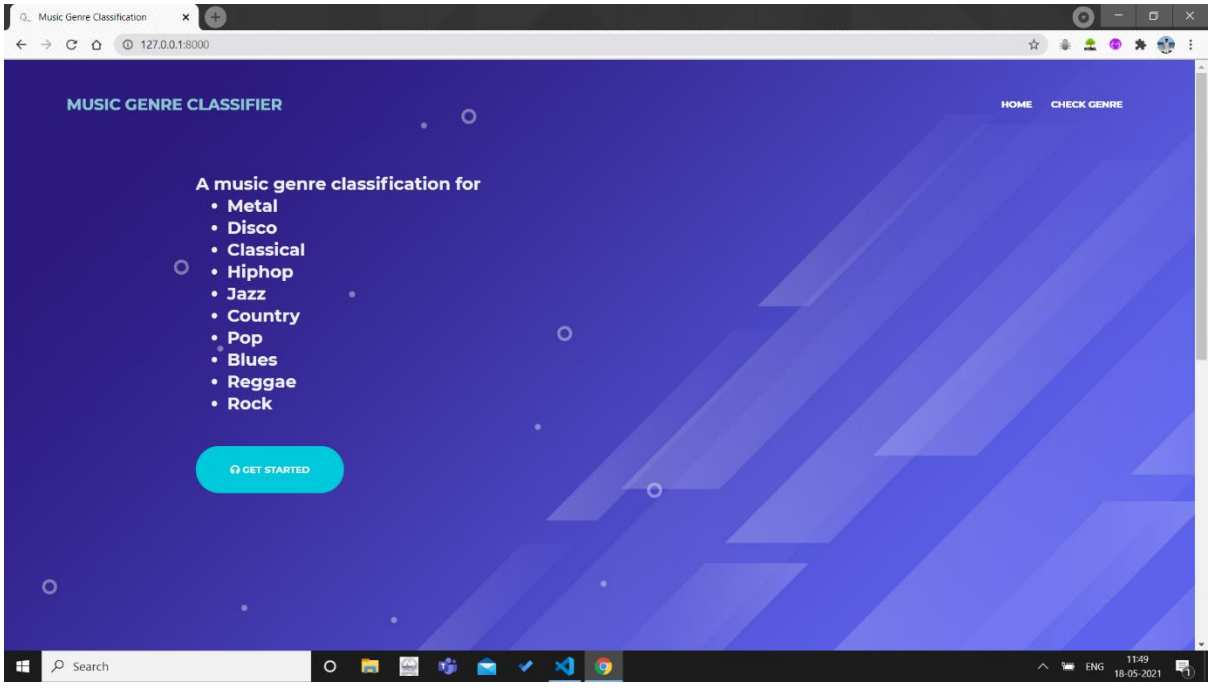
Sample Code from manage.py:

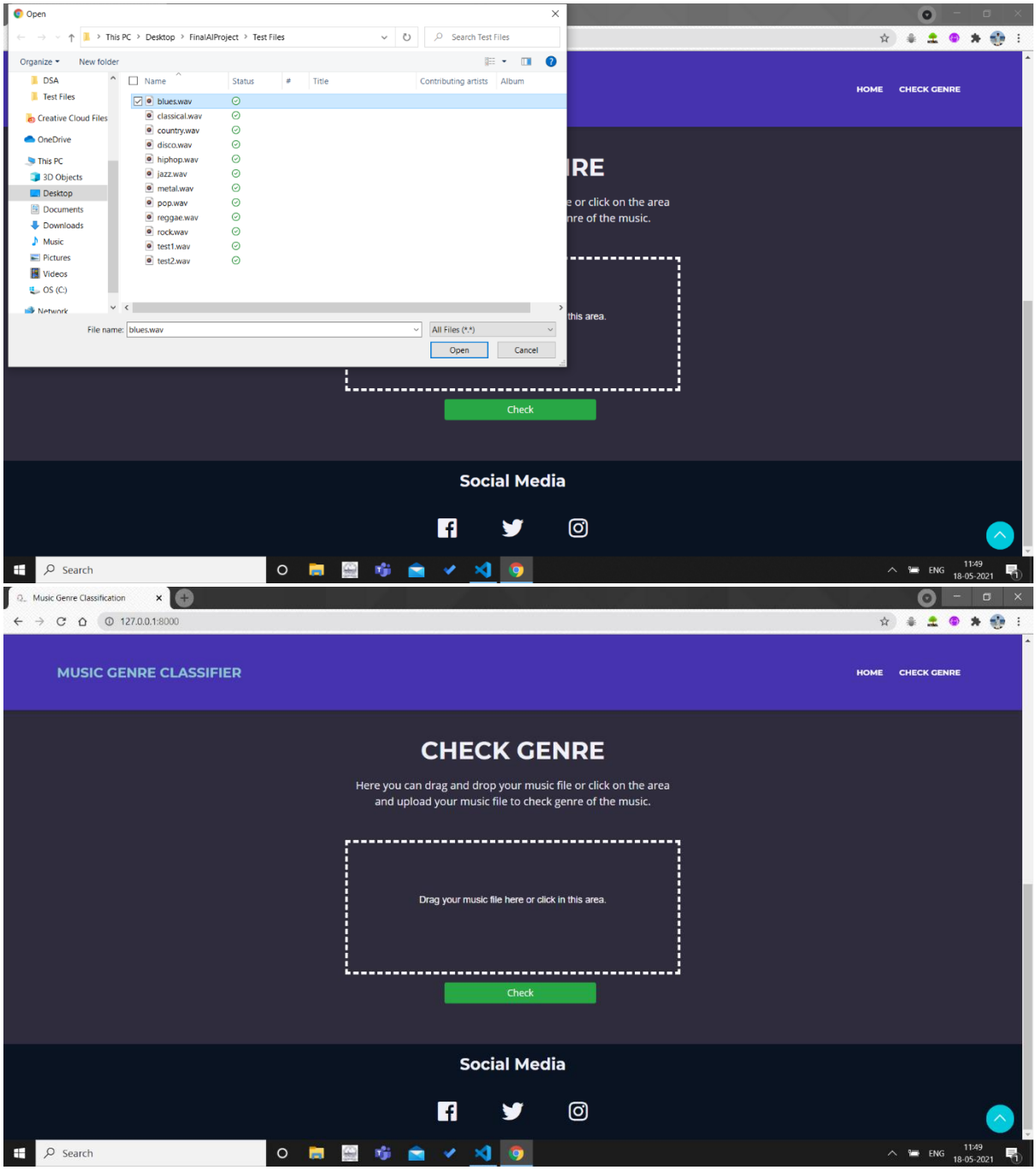
```
import os
import sys

def main():
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'MusicClassification.s
ettings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
```

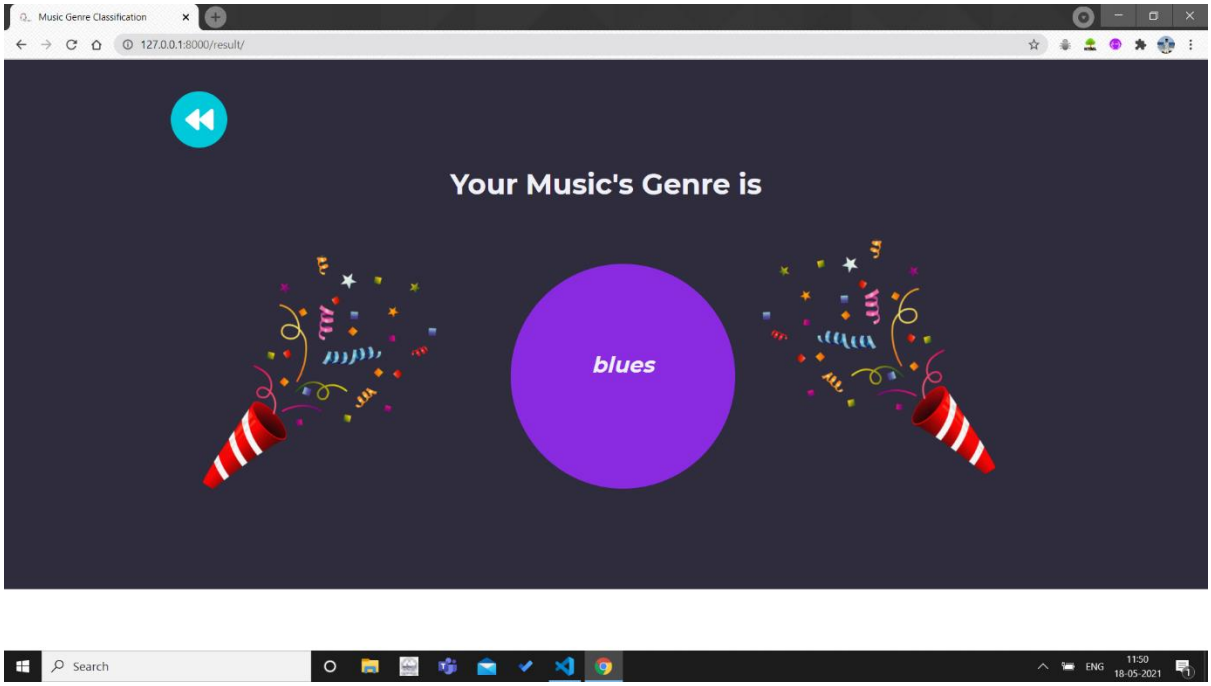
```
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()
```

8.3. Screenshots of all modules:






Open
This PC > Desktop > FinalAIProject > Test Files
Search Test Files
Organize
New folder
Name
Status
Title
Contributing artists
Album
blues.wav
classical.wav
country.wav
disco.wav
hiphop.wav
jazz.wav
metal.wav
pop.wav
reggae.wav
rock.wav
test1.wav
test2.wav
File name: blues.wav
All Files (*.*)
Open
Cancel
HOME
CHECK GENRE
Check
Social Media
Search
ENG
11:49
18-05-2021
Music Genre Classification
127.0.0.1:8000
MUSIC GENRE CLASSIFIER
HOME
CHECK GENRE
CHECK GENRE
Here you can drag and drop your music file or click on the area and upload your music file to check genre of the music.
Drag your music file here or click in this area.
Check
Social Media
Search
ENG
11:49
18-05-2021



9. Conclusion:

In this work, the task of music genre classification is studied using the Audioset data. To solve this dilemma, we suggest two separate approaches. One is using SVM/KNNs, the other is Deep Neural Networks.

Using deep learning, we can achieve the task of music genre classification without the need for hand-crafted features. Specifically, we would like to apply various deep learning algorithms to classify music genres and study their performances.

10. References:

[1] Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hin- ton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. pages 1097–1105.

[3] Andrew Y Ng. 2004. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*. ACM, page 78.

[4] Chadawan Ittichaichareon, Siwat Suksri, and Thaweesak Yingthawornsuk. 2012. Speech recognition using mfcc. In International Conference on Computer Graphics, Simulation and Modeling (ICGSM'2012) July. pages 28–29.

[5] Chanwoo Kim and Richard M Stern. 2012. Power- normalized cepstral coefficients (pncc) for robust speech recognition. In *Acoustics, Speech and Sig- nal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, pages 4101–4104.

[6] Corinna Cortes and Vladimir Vapnik. 1995. Supportvector networks. Machine learning 20(3):273–297.

[7] Dan Ellis. 2007. Chroma feature analysis and synthe- sis. *Resources of Laboratory for the Recognition and Organization of Speech and Audio-LabROSA.*

[8] Dan-Ning Jiang, Lie Lu, Hong-Jiang Zhang, Jian-Hua Tao, and Lian-Hong Cai. 2002. Music type classification by spectral contrast feature. In *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on*. IEEE, volume 1, pages 113–116.

[9] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

[10].E Zwicker and H Fastl. 1999. Psychoacoustics *facts and models*.

[11] Fabien Gouyon, Franc¸ois Pachet, Olivier Delerue, et al. 2000. On the use of zero-crossing rate for an ap- plication of classification of percussive sounds. In *Proceedings of the COST G-6 conference on Digital Audio Effects (DAFX-00), Verona, Italy*.

[12] .George Tzanetakis and Perry Cook. 2002. Musical genre classification of audio signals. *IEEE Trans- actions on speech and audio processing* 10(5):293– 302.

[13] Hagen Soltau, Tanja Schultz, Martin Westphal, and Alex Waibel. 1998. Recognition of music types. In *Acoustics, Speech and Signal Processing, 1998. Pro- ceedings of the 1998 IEEE International Conference on*. IEEE, volume 2, pages 1137–1140.

[14] Jerome H Friedman. 2001. Greedy function approx- imation: a gradient boosting machine. *Annals of statistics* pages 1189–1232.

[15] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio set: An ontology and human-labeled dataset for audio events. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, pages 776–780.

[16] Nicolas Scaringella and Giorgio Zoia. 2005. On the modeling of time information for automatic genre recognition systems in audio signals. In *ISMIR*. pages 666–671.

[17] Leo Breiman. 1996. Bagging predictors. *Machine learning* 24(2):123–140.

[18] Leo Breiman. 2001. Random forests. *Machine learn- ing* 45(1):5–32.

[19] Lonce Wyse. 2017. Audio spectrogram representations for processing with convolutional neural networks. *arXiv preprint arXiv:1706.09559.*

[20] Loris Nanni, Yandre MG Costa, Alessandra Lumini, Moo Young Kim, and Seung Ryul Baek. 2016. Combining visual and acoustic features for music genre classification. *Expert Systems with Applica- tions* 45:108–117.

[21] Michael I Mandel and Dan Ellis. 2005. Song-level fea- tures and support vector machines for music classi- fication. In *ISMIR*. volume 2005, pages 594–599.

[22] Nicolas Scaringella and Giorgio Zoia. 2005. On the modeling of time information for automatic genre recognition systems in audio signals. In *ISMIR*. pages 666–671.

[23] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15(1):1929–1958.

[24] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. 2014. Convolutional neural networks for speech recogni- tion. *IEEE/ACM Transactions on audio, speech, and language processing* 22(10):1533–1545.

[25] Peter Grosche, Meinard Mu¨ller, and Frank Kurth. 2010. Cyclic tempograma mid-level tempo representation for musicsignals. In *Acoustics Speech and Sig- nal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, pages 5522–5525.

[26] Steven B Davis and Paul Mermelstein. 1990. Compar- ison of parametric representations for monosyllabic word recognition in continuously spoken sentences. In *Readings in speech recognition*, Elsevier, pages 65–74.

[27] Suramya Tomar. 2006. Converting video formats with ffmpeg. *Linux Journal* 2006(146):10.

[28] Thomas Lidy and Alexander Schindler. 2016. Parallel convolutional neural networks for music genre and mood classification. *MIREX2016* .

[29] Thomas Lidy and Andreas Rauber. 2005. Evaluation of feature extractors and psycho-acoustic transfor- mations for music genre classification. In *ISMIR*. pages 34–41.

[30] Tom LH Li, Antoni B Chan, and A Chun. 2010. Auto- matic musical pattern feature extraction using con- volutional neural network. In *Proc. Int. Conf. Data Mining and Applications*.

[31] Yali Amit and Donald Geman. 1997. Shape quantiza- tion and recognition with randomized trees. *Neural computation* 9(7):1545–1588.