Dissertation on

"Performance Comparison of Machine Learning Techniques used for Music Genre Classification"

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology
in
Computer Science & Engineering

Submitted by:

Sandeep K Mysore 01FB15ECS262

Under the guidance of

Internal Guide
Raghu B.A. Rao
Associate Professor,
PES University

External Guide
Name of the Guide
Designation,
Company Name

January – May 2019

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)

100ft Ring Road, Bengaluru – 560 085, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

FACULTY OF ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled

'Performance Comparison of Machine Learning Techniques used for Music Genre Classification'

is a bonafide work carried out by

Sandeep K Mysore 01FB15ECS262

In partial fulfilment for the completion of eighth semester project work in the Program of Study Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period Jan. 2019 – May. 2019. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 8th semester academic requirements in respect of project work.

Signature	Signature	Signature
Raghu B.A. Rao	Dr. Shylaja S S	Dr. B K Keshavan
Associate Professor	Chairperson	Dean of Faculty

External Viva

Name of the Examiners	**Signature with Date**
1. _____	_____
2. _____	_____

DECLARATION

I hereby declare that the project entitled **"Performance Comparison of Machine Learning Techniques used for Music Genre Classification"** has been carried out by me under the guidance of Raghu B.A. Rao, Associate Professor and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester January – May 2019. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

01FB15ECS262 Sandeep K Mysore

ACKNOWLEDGEMENT

I would like to express my gratitude to Associate Professor, Raghu B.A. Rao, Dept. of Computer Science, PES University, for his continuous guidance, assistance and encouragement throughout the development of this project.

I am grateful to the project coordinators, Prof. Preeth, for organizing, managing and helping out with the entire process.

I take this opportunity to thank Dr. Shylaja S S, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department. I would like to thank Dr. B.K. Keshavan, Dean of Faculty, PES University for his help.

I am deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. K.N.B. Murthy, Vice-Chancellor and PES University for providing to me various opportunities and enlightenment every step of the way. Finally, this project could not have been completed without the continual support and encouragement I have received from my mother.

ABSTRACT

Music Genre Classification is being used in multiple Music Streaming websites and applications to provide the users with means to understand more about the characteristics of the music they are listening to and to provide genre wise categorical recommendations. Yet, this is a very daunting task in the Music Information Retrieval (MIR) domain.

This project demonstrates the performance, in terms of classification accuracy, of various Machine Learning and Deep Learning Algorithms for the purpose of classifying music based on genre. K-Nearest Neighbors (KNN), Logistic Regression, Support Vector Machine (SVM) are the Machine Learning algorithms being compared in this project along with Convolutional Neural Network (CNN) which is a Deep Learning algorithm.

The algorithms are also trained on the features extracted from the GTZAN dataset which consists of audio clips for 10 different genres. The feature extraction methods used in this project are Fast-Fourier Transforms (FFT) and Mel-Frequency Cepstral Coefficients (MFCC).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

1.1 Overview

The task at hand is to classify music based on genres using different Machine Learning algorithms to compare their performances in terms of classification accuracy. This project comes under the Music Information Retrieval domain as well as the Signal Processing domain, not to mention Artificial Intelligence.

As this is an ongoing area of research, there are multiple methods of implementing the Machine Learning Algorithms as well as for extracting features.

1.2 Scope

The scope of this project includes analysing the audio signals of the GTZAN dataset, which is in .au format, convert it to a more suitable .wav format and extract features from it. For feature extraction we have used Fast Fourier Transforms (FFT) and Mel Frequency Cepstral Coefficients (MFCC) mainly because FFT has been around for some time and MFCC is a more modern method. This provides a contrast, in development of algorithms with respect to time.

The ML algorithms chosen for classification are Logistic Regression, KNN and SVM. CNN, a deep learning algorithm is also implemented for this purpose. The outcome will be classification accuracies of each of the feature-model pairs.

1.3 Objective

The objective is to compare the accuracies of different Machine Learning algorithms which include LR, SVM, KNN and deep learning algorithm CNN. The Confusion matrices are also obtained to study the performance effectively.

The dataset used for this purpose is the GTZAN dataset. Another dataset is used for Genre classification of Classical music.

1.4 Outcomes

The overall outcome of this project is to compare the performance of different ML algorithms based on their accuracies on the GTZAN dataset and to carry out the same on the **"**A Database for Indian Classical Music" dataset which consists of different ragas.

Chapter-2

Research Background

2.1 Literature Survey

2.1.1 "An analysis of the GTZAN Music Genre Dataset" by Bob L Strum

2.1.1.1 Introduction

In their work [1] automatic music genre recognition, Tzanetakis et al. created a dataset (GTZAN) of 1000 music excerpts of 30 seconds duration with 100 examples in each of 10 different music genres: Blues, Classical, Country, Disco, Hip Hop, Jazz, Metal, Popular, Reggae, and Rock. Its availability has made possible much work exploring the challenges of making machines recognize something as complex, abstract, and often argued arbitrary, as musical genre.

From their analysis of the 1000 excerpts in GTZAN, they found: exact replicas (including one that is in two classes), 22 excerpts from the same recording,13 versions (same music, different recording), and 44 conspicuous and 64 contentious mislabeling. we also find significantly large sets of excerpts by the same artist e.g., 35 excerpts labeled Reggae are of Bob Marley, 24 labeled pop are of Britney Spears, and so on.

2.1.1.2 Methodology

In this paper, the problem of repetition at a variety of specificities is considered. From high to low specificity, they are: excerpts are exactly the same; excerpts come from same recording (displaced in time, time-stretched, pitch-shifted, etc.); excerpts are of the same song (versions or covers); excerpts are by the same artist. The most highly-specific repetition of these is exact, and can be found by a method having high specificity, e.g., fingerprinting.

The second problem is mislabeling, which is considered in two categories: conspicuous and contentious. A mislabeling conspicuous is considered when there are clear musicological criteria and sociological evidence to argue against it. Musicological indicators of genre are those characteristics specific to a kind of music that establish it as one or more kinds of music, and that distinguish it from other kinds. Examples include: composition, instrumentation, meter, rhythm, tempo, harmony and melody, playing style, lyrical structure, subject material, etc. Sociological indicators of genre are how music listeners identify the music, e.g., through tags applied to their music collections.

The third problem is distortion. Though Tzanetakis et al. purposely created the dataset to have a variety of fidelities, he found errors such as static, and digital clipping and skipping.

2.1.2 "Million Song Dataset" by Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman and Paul Lamere

2.1.2.1 Introduction

This paper [2] talks about the Million Song Dataset (MSD) which is an attempt to help researchers by providing a large-scale dataset. It contains metadata and audio analysis for a

million songs that were legally available to The Echo Nest. The paper describes the main purposes of the dataset as follows:

- encouraging research on algorithms that scale to commercial sizes;
- providing a reference dataset for research;
- shortcut substitute to creating a large dataset with The Echo Nest's API;
- to help new researchers get started in the MIR field.

The paper also lists some of the advantages to creating a large dataset:

- A large dataset helps reveal problems with algorithm scaling that may not be so obvious or pressing when tested on small sets, but which are critical to real world deployment.
- Various kinds of relatively-rare occurrences or patterns may not be perceptible in small datasets, but may lead to exciting discoveries from large collections.
- A large dataset can encompass various more specialized subsets. By grouping all subsets within a single entity, we can have standardized data fields, features, etc.
- A single, versatile, freely-accessible dataset greatly promotes direct comparisons and interchange of ideas and results.

2.1.2.2 Content

The MSD consists of audio features and metadata for a million modern popular music tracks. It contains:

- 280 GB of data
- 1, 000, 000 songs/files
- 44, 745 unique artists
- 7, 643 unique terms (Echo Nest tags)
- 2, 321 unique musicbrainz tags

- 43, 943 artists
- 2, 201, 916 asymmetric similarity relationships
- 515, 576 dated tracks starting from 1922

The data is stored using HDF5 format 2 to efficiently handle the heterogeneous types of information such as audio features in variable array lengths, names as strings, longitude/ latitude, similar artists, etc. Each song is described by a single file.

2.3 "Music Genre Classification using Deep Neural Networks" by G. Jawaherlalnehru, S. Jothilakshmi

In this paper, [3] a deep learning approach is proposed for automatic genre identification and it is evaluated with GENRE Dataset. It was concluded that DNN with MFCC features works well better than the other system. The accuracy obtained was 97.8% in the proposed system. DNN is the best classifier model according to this study. The future work includes increasing databases with other feature techniques. Experimenting larger amounts of data (i.e., Million Song Dataset) and developing new features which can extract the meaningful information from the audio signals and use more feature sets such as melodic characteristics.

2.4 "A Study on Different Music Genre Classification Methods" by Alif Noushad, Albin Paul, Anjana Mukesh, Ebin B Plackal, Mohan T D and Anjali S

In this paper [4], an extensive survey on various Music Genre Classification techniques has been depicted. In the above study it was inferred that using MFCC data values gives better results overall than using FFT values. PLP and MFCC are derived on the concept of logarithmically spaced filterbank, clubbed with the concept of human auditory system hence has better performance.

The Simpler algorithms such as Logistic Regression and K Nearest Neighbors did fairly well compared to superior algorithms such as Recurrent Neural Networks and Support Vector Machines. Pattern matching is efficiently done using Recurrent Neural Networks. All others techniques discussed above failed to produce pattern matching as efficient as Recurrent Neural Networks. Performance of technique like RASTA has been increased by combining RASTA with PLP hence ensuring better performance ratio.

2.5 "Automatic Music Genre Classification of Audio Signals with Machine Learning Approaches" by Y.M.D. Chathuranga and K.L. Jayaratne

In this paper [5], the problem of musical genre classification from audio signals is addressed. The researches in this area are very concerned about the classification accuracy. In this research it is verified that it is possible to improve the classification accuracy by using machine learning algorithms and different types of audio features together. An alternative approach for music genre classification based on classifier ensemble techniques is presented and GTZAN dataset and ISMIR2004 dataset are evaluated. Results showed that use of late fusion methods can improve the classification results in a more robust way than using early fusion approaches.

AdaBoost boosting algorithm performed well when the classifiers are weak but using their proposed features SVM with polynomial kernel function acted as a strong base learner in AdaBoost, so its performance of the SVM classifier cannot improve using boosting method.

They have also used a filtering and wrapping algorithm for feature selection in order to create a reduced feature vector.

Filtering approach provides the same accuracy as the feature vector containing all features but with a compact representation and wrapper approach provides both high accuracy and compact representation. SVM with a polynomial kernel as an individual classifier was used and using ensemble classifier, music genre classification accuracy has been obtained 78% and 81% on GTZAN dataset and ISMIR2004 respectively.

Chapter-3

Methodology

3.1 Proposed Approach

The task at hand is to classify music based on genres using different Machine Learning algorithms to compare their performances in terms of classification accuracy. This project comes under the Music Information Retrieval domain as well as the Signal Processing domain, not to mention Artificial Intelligence.

As this is an ongoing area of research, there are multiple methods of implementing the Machine Learning Algorithms as well as for extracting features.

3.1.1 What is the proposed approach?

The proposed approach is to analyze the audio signals of the GTZAN dataset, which is in .au format, convert it to a more suitable .wav format and extract features from it. For feature extraction we have used Fast Fourier Transforms (FFT) and Mel Frequency Cepstral Coefficients (MFCC) mainly because FFT has been around for some time and MFCC is a more modern method. This provides a contrast, in development of algorithms with respect to time.

The ML algorithms chosen for classification are CNN, KNN and SVM. CNN, a deep learning algorithm is also implemented for this purpose. The outcome will be classification accuracies of each of the feature-model pairs.

3.1.2 What are the other approaches?

- Feature extraction can be done from Audio meta-data instead of audio signals. For this, The Million Song Dataset is perfect.
- Other feature extraction methods like LPCC, LFCC, etc. gives different feature vector sets which can be used to train the model.
- Machine Learning algorithms like Random Forest or Artificial Neural Network (ANN) can be implemented for this purpose.
- The combinations and possibilities of implementing this are endless and is therefore a hot topic for research.

3.2 High Level System Architecture

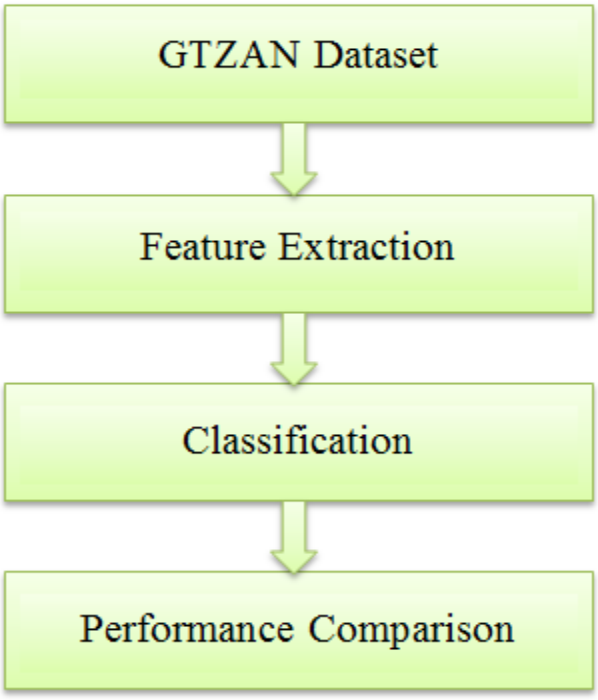

Fig 3.2 High Level System Architecture

- *Phase 1 : Audio Dataset*

We have used the GTZAN dataset [6] from the MARYSAS website which contains 6 music genres, each having 100 audio clips in .au format. The different genres in this dataset are - blues, classical, country, disco, pop, and rock. Each audio clip has a length of 30 seconds, are 22050Hz Mono 16-bit files. The dataset incorporates samples from various sources like CDs, radios, microphone recordings etc.

Another dataset of Classical music called "A Database for Indian Classical Music" [7] is used. This dataset consists of Thaat, Ragas and their Midi Files. It has 10 Thaat examples.

- *Phase 2 : Feature Extraction*

The process of converting an audio signal into a sequence of feature vectors by extracting relevant characteristics enclosed the dataset is called feature extraction. It provides a compact representation by reducing the redundant information in the audio signal.

- *Phase 3 : Classification*

Classification is the process of applying various Machine Learning algorithms on the extracted features so that it can predict the music genre for an unknown audio file. KNN, SVM, CNN, and Logistic Regression are the algorithms used for this purpose.

- *Phase 4 : Performance Comparison*

The performance of the algorithms mentioned in the classification phase is analyzed and their classification accuracies are reported. The Confusion matrix is plotted for all the algorithms.

3.3 Low Level System Architecture

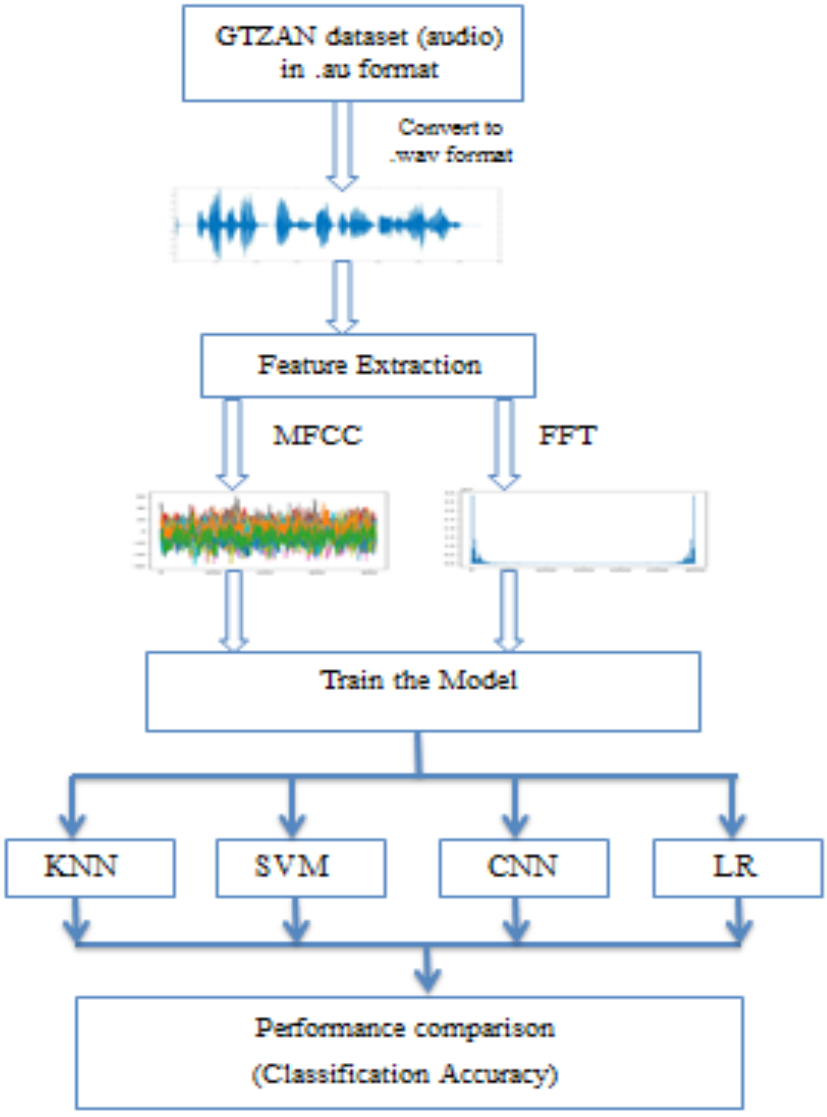

Fig 3.3 Low Level System Architecture

3.3.1 Modules

- *Data Preprocessing*

The preprocessing stage involves converting the audio files in the GTZAN dataset from .au format to .wav format so that it is compatible to python's wave module for reading the audio files. There are various methods available for converting the audio files. The open source SoX [8] utility was used for this conversion.
Before using SoX utility, pydub library's audio segment function was used to perform the conversion but many errors were encountered and therefore I shifted to the SoX method.

- *Feature Extraction*

To classify the audio clips and extract the relevant features two methods were used - Mel-Frequency Cepstral Coefficients (MFCC) and Fast Fourier Transforms (FFT).

1. Fast Fourier Transforms (FFT).

In various audio classification systems, for the classification of audio signals a spectrogram or other joint time-frequency representations are used. A feature vector is obtained in these features by applying the Fast Fourier Transform (FFT) to the windowed audio signals. In our proposed method, the spectrogram is used as the starting point of the feature extraction process. In the first step, the audio signal is partitioned using a Hamming window function. In the next step, the short term features are extracted from

the audio signal segments. Then, the amplitudes of the FFT coefficients of each frame are normalized by dividing them by their maximum value. Now we have short-term feature vectors. In the fourth step, a new feature vector is formed in which each component is the sum of the normalized coefficients of the corresponding frame. As already mentioned, the audio signals are not stationary over a relatively long time interval. However, in the constructed feature space, the statistical properties of the resulting data become almost constant and can be considered as stationary. In the next step, the FFT of the vector composed of the feature samples of this stationary data is computed to extract long-term feature. This second FFT provides a sparse representation. Applying an amplitude filter a sparse feature vector is obtained [9]. In step six, we obtained used samples for music genre classification by random sampling. In this feature extraction process, the number of features is reduced from the number of samples to the number of frames. Moreover, the feature vector is sparse.

2. Mel-Frequency Cepstral Coefficients (MFCC) :

The information of rate of change in spectral bands is known as Cepstrum. In the conventional analysis of time signals, any periodic component (for e.g., echoes) will be shown as sharp peaks in the corresponding frequency spectrum (i.e., Fourier spectrum. This is obtained by applying a Fourier transform on the time signal). This can be seen in the following image.

When the log of the magnitude of this Fourier Transform is taken and then again the spectrum of this log is taken by Cosine Transformation, a peak is observed wherever there is a periodic element in the original spectrum. The resulting spectrum is neither in the frequency domain nor in the time domain as we have applied the transform on the frequency spectrum itself. Hence Bogert et al. [10] called it the Quefrency domain and this spectrum of the log of the spectrum of the time signal was named cepstrum.

Sometimes there is a gap between perception and reality. This is true even in case of audio perception by the human ear. The perception of the difference in sound when it changes from, let us say, 100Hz to 200Hz is far less when compared to the difference perceived when it changes from 1000Hz to 1100Hz even though the actual difference is the same.

Mel scale maps this difference, in the case of audio signal frequencies by implementing the given formula below.

$$\text{Mel}(f) = 2595 \log\left(1 + \frac{f}{700}\right)$$

Fig 3.3.1.1 Formula for Mel Frequency

The envelope of the time power spectrum of an audio signal represents the MFCC coefficients that make up the Mel-frequency cepstrum. A step-wise summary of how we arrived at MFCC is shown in the following diagram:

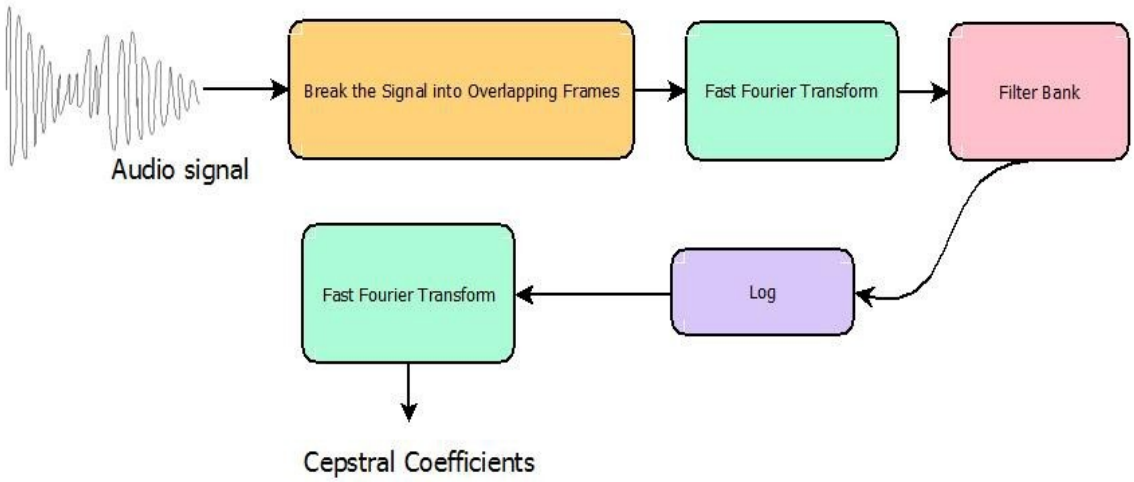

Fig 3.3.1.2 Step-wise summary of MFCC

Here, Filter Bank refers to the mel filters (converting to mel scale) and Cepstral Coefficients are nothing but MFCCs.

- *Classification*

Once the feature sets are extracted, we train different Machine Learning algorithms on the this set. Following are the different algorithms that were used -

- K Nearest Neighbours:

KNN is a nonparametric classifier. The error of KNN being asymptotically at most twice as large as the Bayesian error rate has been proven. KNN has been successfully applied in many analysis problems. The basic idea is to allow a small number of neighbors to influence the decision on a point.

```
knn_classifier = KNeighborsClassifier()
knn_classifier.fit(X_train, y_train)
knn_predictions = knn_classifier.predict(X_test)
knn_accuracy = accuracy_score(y_test, knn_predictions)
knn_cm = confusion_matrix(y_test, knn_predictions)
```

Fig 3.3.1.2.2 KNN Code Snippet

- Support Vector Machine:

A machine learning technique which is based on the principle of structure risk minimization is support vector machines. It has many applications in the area of pattern recognition [11]. A linear model based on the support vectors is constructed in order to estimate decision function. If the training data are linearly separable, then SVM finds the optimal hyper plane that separates the data without error [12].

SVM maps the input patterns through a non-linear mapping into higher dimension feature space. A linear SVM is used to classify the data sets [13] which can be separated linearly. The support vectors are the patterns lying on the margins which are maximized.

The support vectors are the training patterns that are transformed and are equally close to hyper plane of separation. The support vectors are the training samples that define the optimal hyper plane and are the most difficult patterns to classify [14] and they are the patterns of the classification task which are the most informative.

```
clf = svm.SVC(kernel='linear', C=1).fit(X_train, y_train)
svm_predictions = clf.predict(X_test)
svm_accuracy = accuracy_score(y_test, svm_predictions)
svm_cm = confusion_matrix(y_test, svm_predictions)
```

Fig 3.3.1.2.3 SVM Code Snippet

- Logistic Regression:

  This classifier is mostly used for binary classification tasks. For this multi-class classification task, the LR is implemented as a one-vs-rest method i.e., 7 separate

binary classifiers are trained. During test time, the class with the highest probability from among the 7 classifiers is chosen as the predicted class.

```
logistic_classifier = linear_model.logistic.LogisticRegression()
logistic_classifier.fit(X_train, y_train)
logistic_predictions = logistic_classifier.predict(X_test)
logistic_accuracy = accuracy_score(y_test, logistic_predictions)
logistic_cm = confusion_matrix(y_test, logistic_predictions)
```

Fig 3.3.1.2.1 Logistic Regression Code Snippet

- Convolutional Neural Network:

In deep learning, a class of deep neural networks that are applied to analyzing visual imagery are called convolutional neural network (CNN, or ConvNet) [15].

CNNs are nothing but regularized versions of multilayer perceptrons. Multilayer perceptrons refer to fully connected networks in which each neuron in one layer is connected to all neurons in the next layer. The "fully-connectedness" of these networks makes them prone to overfitting data. Different ways of regularization includes adding of magnitude measurement of weights to the loss function. However, CNNs take advantage of the hierarchical pattern in data and assemble more complex patterns using smaller and simpler patterns. Therefore, CNNs are on the lower extreme on the scale of connectedness and complexity.

The implementation of CNN consists of running it for 20 epochs with a 0.2 Validation_split.

- *GUI:*

The project is demonstrated with a basic web UI which will contain Javascript Buttons which in turn calls the scripts which will extract features, train the model and output the classification accuracy. The frontend part of the UI is built using Bootstrap, Javascript, Html5 and Css3.

Chapter-4

Environment Requirements

4.1 Hardware Requirements

Since we are using Tensorflow and training Deep Neural Networks, it is recommended to have the following Hardware Requirements:

- Processor: Intel® Core™ i7 processor that has 2.60 or 2.59 GHz Clock Cycle with 2 cores and 2 threads per core.

The Minimum Hardware requirements of the Scanner are:

- Operating System: The Web application Vulnerability Scanner can be run on all the hardware devices that can host Ubuntu Operating system with 16.04 and above.
- Processors: An intel Atom® processor or Intel® Core™ i5 processor with 2 GHz clock speed.
- Disk Space: A Disk Space of 1 GB
- RAM: A RAM of 2GB
- Network: Wired or Wireless active internet connection

4.2 Software Requirements

The software requirements are:

- Python
  - Version: 2.7
  - Source: https://www.python.org/

- Pycharm IDE
  - Description: Python IDE for professional developers
  - Version: 3
  - Source: https://www.jetbrains.com/pycharm/

Packages	Version
Tensorflow	1.13.1
Keras	2.2.4
Numpy	1.16.2
Pip	19.0.3
Conda	4.6.14
Python-speech-features	0.6
Scikit-learn	0.20.3
Scipy	1.2.1
Sox	1.3.7
Librosa	0.6.3

Fig 4.2 Package and their Versions

4.3 Data Requirement

We have used the GTZAN dataset [9] from the MARYSAS website which contains 6 music genres, each having 100 audio clips in .au format. The different genres in this dataset are:

1. Blues
2. Classical
3. Country
4. Disco
5. Pop
6. Rock

Each audio clip has a length of 30 seconds, are 22050Hz Mono 16-bit files. The dataset incorporates samples from various sources like CDs, radios, microphone recordings etc.

Chapter-5

Demonstration of Outcome

- The project is demonstrated with a basic web UI which will contain Javascript Buttons which in turn calls the scripts which will extract features, train the model and output the classification accuracy.
- The model is trained by using various Machine Learning algorithms which are KNN, SVM and Logistic Regression and Deep Learning algorithm CNN.
- The output will be the accuracy of the feature - model pair and the confusion matrix of the model.
- The frontend part of the UI is built using Bootstrap, Javascript, Html5 and Css3.

Chapter-6

Proposed Approach

The proposed approach is to convert GTZAN dataset, which is in .au format, to a more suitable .wav format and extract features from it. For feature extraction we have used Fast Fourier Transforms (FFT) and Mel Frequency Cepstral Coefficients (MFCC) mainly because FFT has been around for some time and MFCC is a more modern method. This provides a contrast, in development of algorithms with respect to time.

The ML algorithms chosen for classification are CNN, KNN and SVM. CNN, a deep learning algorithm is also implemented for this purpose. The outcome will be classification accuracies of each of the feature-model pairs.

The proposed approach:

- Convert the .au audio data from GTZAN dataset using SoX
- Extract Features for FFT and MFCC using Librosa and Scipy Package
- Use of Matplotlib.pyplot to plot confusion matrix and graphs
- Use of sklearn package to train the machine learning models.
- Use of Vgg-16 Architecture to implement CNN.
- Performance Comparision of feature-model pair.
- Use of Indian Classical Music from IIT Hyderabad, instead of Raaga Classification.
- Extract Features of the Indian Classical Dataset
- Train the model on these features to check compatibility.

Chapter-7

Timeline of Approaches

7.1 Initial Approach

7.1.1 Choice of Dataset

- The initial choice of dataset was “The Million Song Dataset”. The main reason for this is it’s fame in this domain. This dataset consists of meta-data instead of audio files.
- The proposed approach was to classify music using the audio file instead of meta-data.
- So, this dataset was later dropped.

7.1.2 Data Preprocessing Methods

- The initial approach for data pre-processing, i.e converting the dataset from .au to .wav format was to use Pydub’s Audio Segment function. But, because of some problems that were encountered, this method was dropped.

7.1.3 Choice of method for Feature Extraction

- There are multiple feature extraction methods LPCC, PLP, Spectral Rolloff, etc.
- Fast-Fourier Transform was the initial chosen approach due to its long-term existence

7.1.4 Method of Implementing CNN

- The initial approach was to use 3 layers run 5 times in a sequential order consisting of Conv2D, MaxPooling2D and Dropout Layers.

```
def createCNN1():
    model = Sequential()
    # Conv Block 1
    model.add(Conv2D(16, kernel_size=(3, 3), padding="same", activation='relu', input_shape=input_shape))
    model.add(MaxPooling2D(2))
    model.add(Dropout(0.1))

    # Conv Block 2
    model.add(Conv2D(32, (3, 3), padding="same", activation='relu'))
    model.add(MaxPooling2D(2))
    model.add(Dropout(0.1))

    # Conv Block 3
    model.add(Conv2D(64, (3, 3), padding="same", activation='relu'))
    model.add(MaxPooling2D(2))
    model.add(Dropout(0.1))

    # Conv Block 4
    model.add(Conv2D(128, (3, 3), padding="same", activation='relu'))
    model.add(MaxPooling2D(2))
    model.add(Dropout(0.1))

    # # Conv Block 5
    model.add(Conv2D(256, (3, 3), padding="same", activation='relu'))
    model.add(MaxPooling2D(2))
    model.add(Dropout(0.1))

    # MLP
    model.add(Flatten())
    model.add(Dense(num_genres, activation='softmax'))

    model.summary()
    model.compile(loss=keras.losses.categorical_crossentropy,
                  optimizer=keras.optimizers.Adam(),
                  metrics=['accuracy'])
    return model
```

Fig 7.1.4 CNN1 snippet

7.2 Final Approach

7.2.1 Choice of Dataset

- GTZAN dataset was finally decided upon.
- This dataset was chosen because of the format of the dataset (.au format). This was inline with the proposed approach.

7.2.2 Data Preprocessing Methods

- SoX or Sound eXchange was the final approach used for data pre-processing.
- It was chosen as it is a very powerful cross-platform package/ utility that can handle multiple audio formats
- It is rightly called the “The Swiss Army Knife” of sound processing programs.

7.2.3 Choice of method for Feature Extraction

- The feature extraction method finally used were both FFT and MFCC
- MFCC was chosen as it is a modernized and improvised feature extraction method which keeps FFT as part of the process.
- The other reason was to provide a contrast between a modern and an older feature extraction method.

7.2.4 Method of Implementing CNN

- The final approach to implementing CNN was to use the VGG-16 Architecture.
- This consists of 16 Layers in total. A combination of 4 Layers in is run sequentially 4 times. The layers consist of Conv2D layer applied twice, followed by a MaxPooling2D layer to reduce spatial dimensions followed by a dropout layer.

```
def createCNN2():
    model = Sequential()
    # Conv Block 1
    model.add(Conv2D(128, kernel_size=(3, 3), padding="same", activation='relu', input_shape=input_shape))
    model.add(Conv2D(128, (3, 3), padding="same", activation='relu'))
    model.add(MaxPooling2D(2))
    model.add(Dropout(0.5))

    model.add(Conv2D(64, kernel_size=(3, 3), padding="same", activation='relu'))
    model.add(Conv2D(64, (3, 3), padding="same", activation='relu'))
    model.add(MaxPooling2D(2))
    model.add(Dropout(0.5))

    model.add(Conv2D(32, kernel_size=(3, 3), padding="same", activation='relu'))
    model.add(Conv2D(32, (3, 3), padding="same", activation='relu'))
    model.add(MaxPooling2D(2))
    model.add(Dropout(0.5))

    model.add(Conv2D(16, kernel_size=(3, 3), padding="same", activation='relu'))
    model.add(Conv2D(16, (3, 3), padding="same", activation='relu'))
    model.add(MaxPooling2D(2))
    model.add(Dropout(0.5))

    model.add(Flatten())
    model.add(Dense(num_genres, activation='softmax'))

    model.summary()
    model.compile(loss=keras.losses.categorical_crossentropy,
                  optimizer=keras.optimizers.Adam(),
                  metrics=['accuracy'])
    return model
```

Fig 7.2.4 CNN2 snippet

Chapter-8

Results

- FFT-Logistic Regression pair :

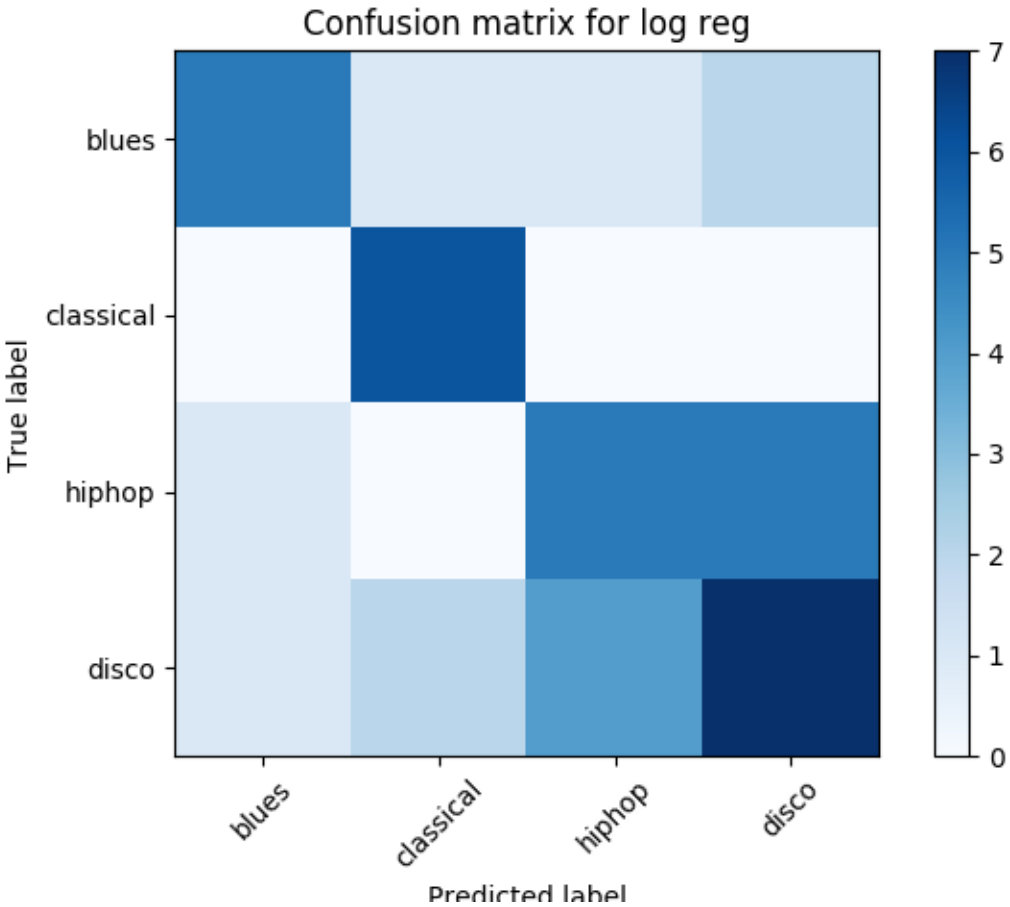

Fig 8.1 Confusion matrix for FFT-Logistic Regression

- FFT-SVM pair :

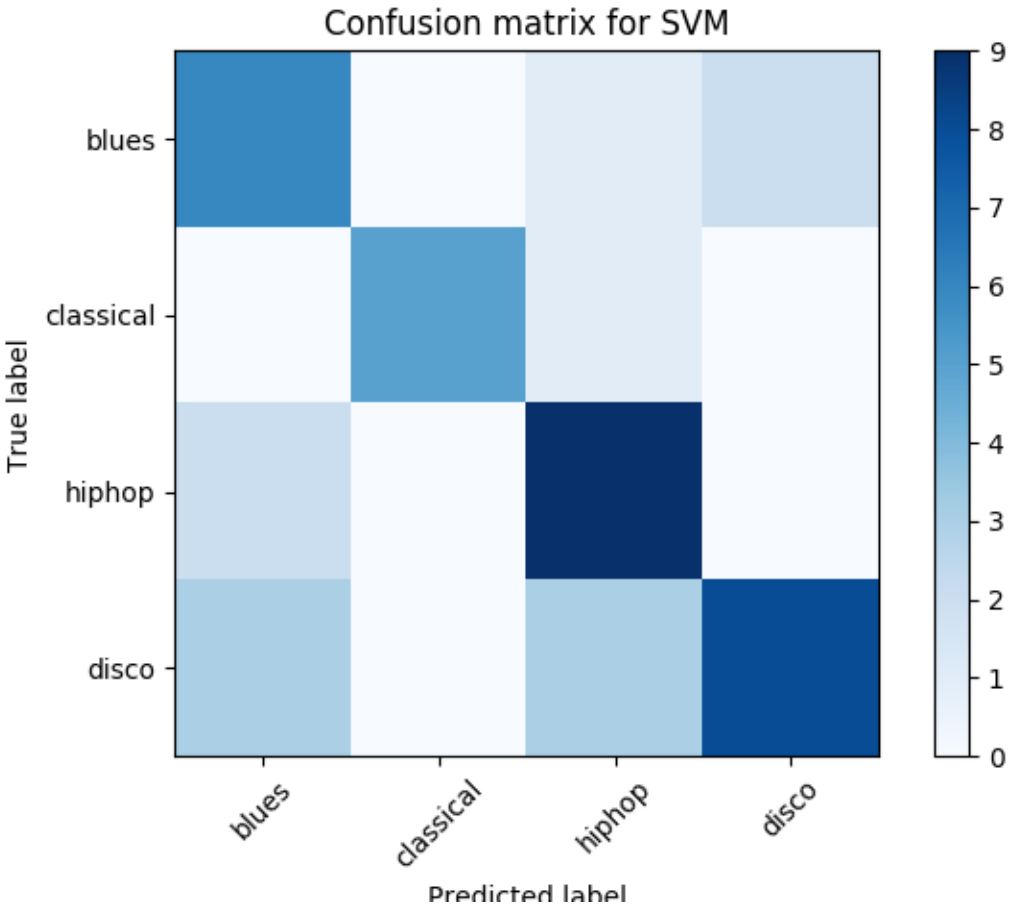

Fig 8.2 FFT-SVM pair

- FFT-KNN pair:

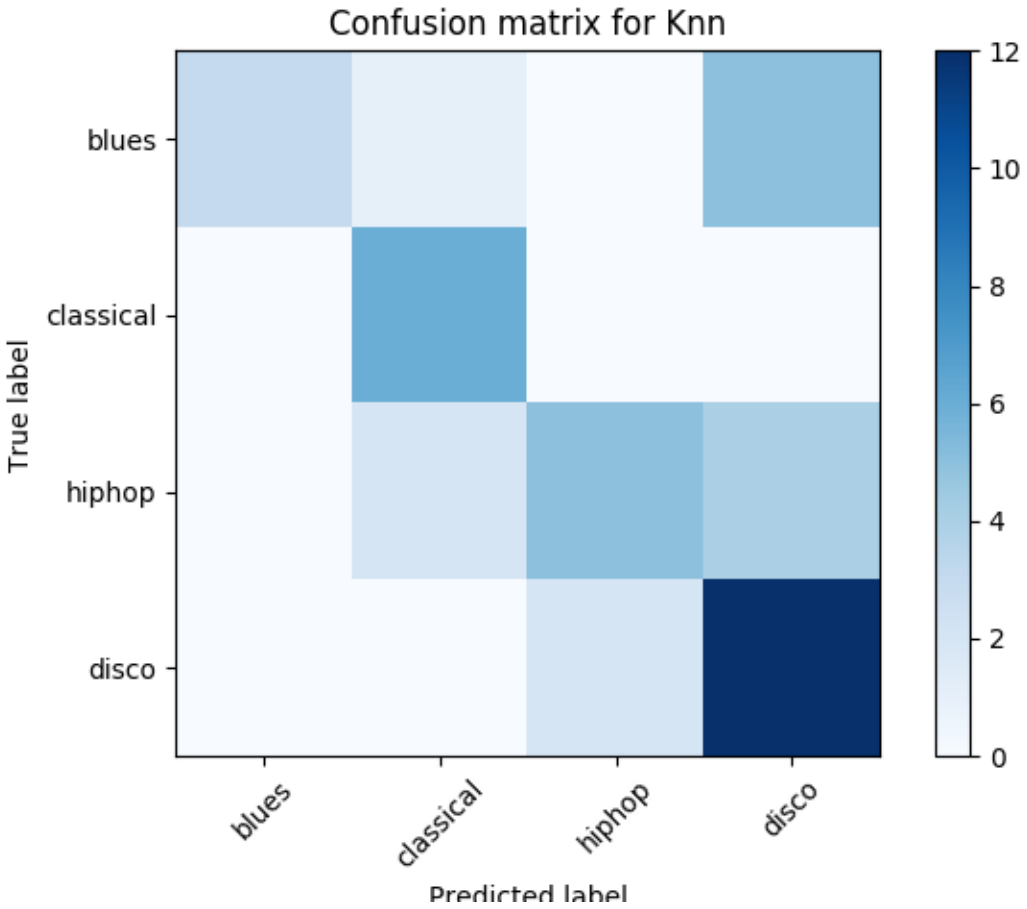

Fig 8.3 FFT-KNN pair

- MFCC-Logistic Regression:

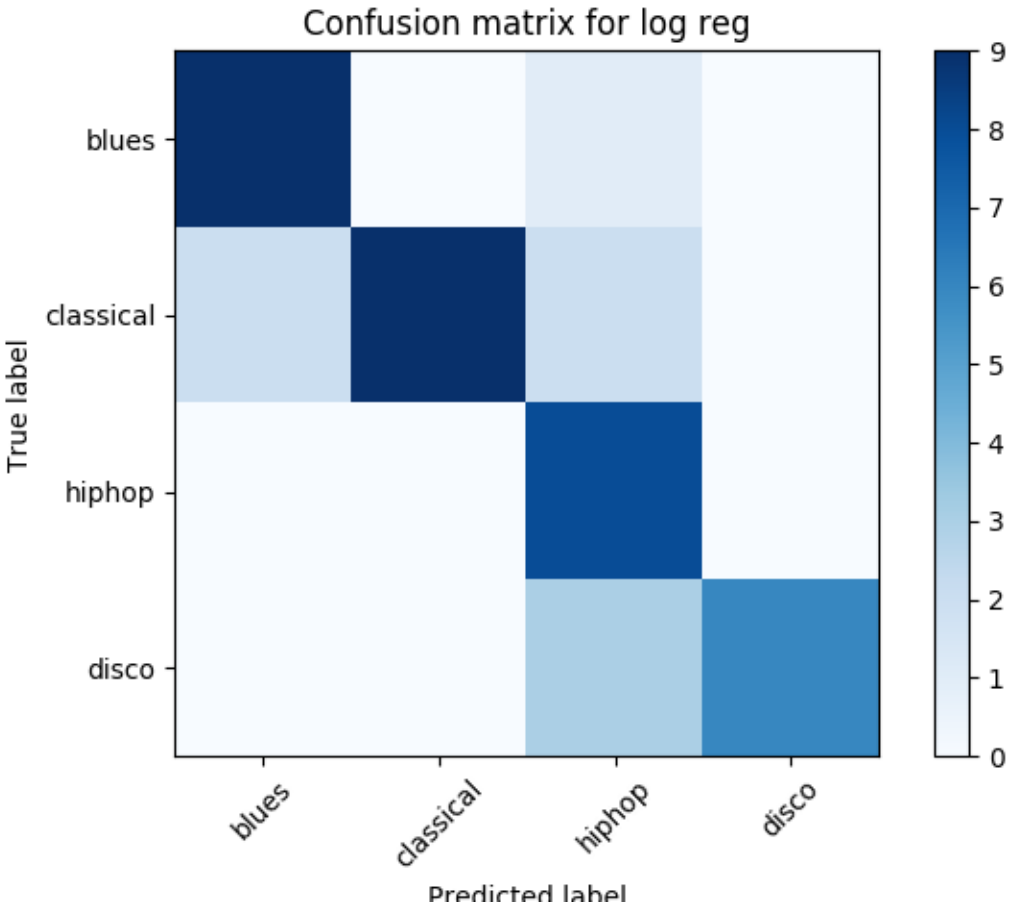

Fig 8.4 MFCC LR pair

- MFCC-SVM:

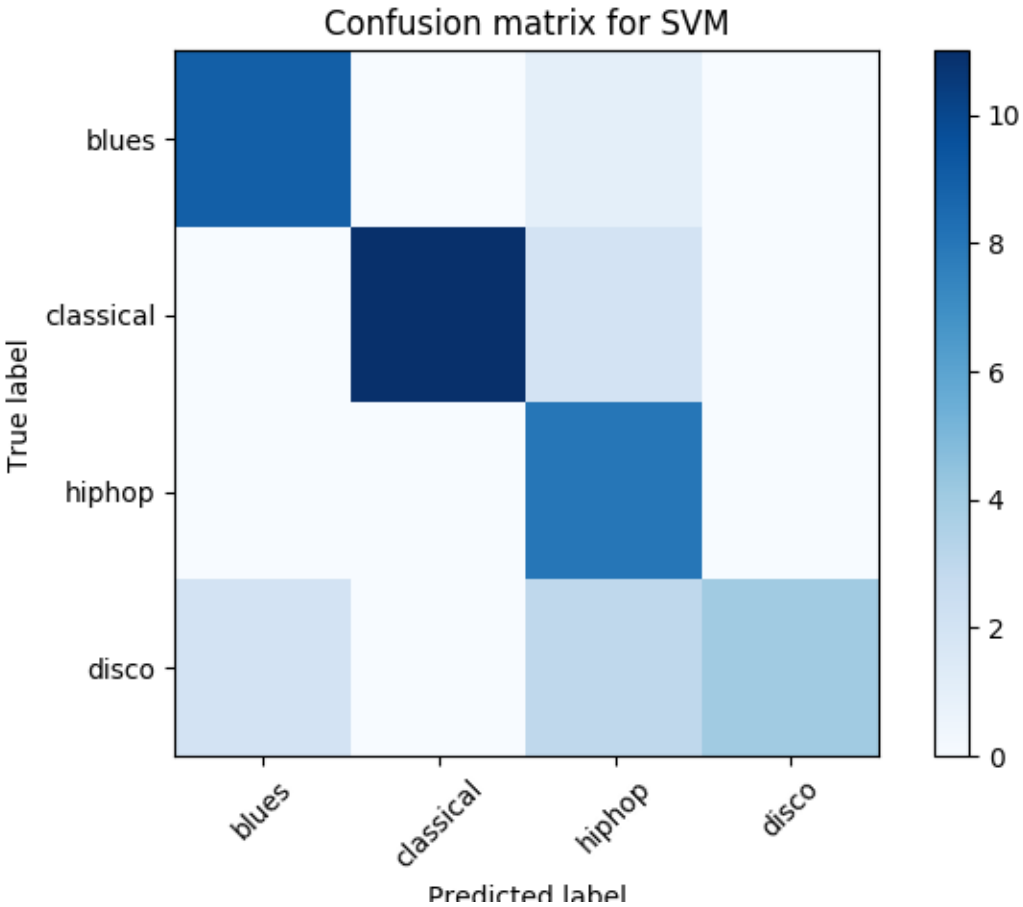

Fig 8.5 MFCC-SVM pair

- MFCC-KNN:

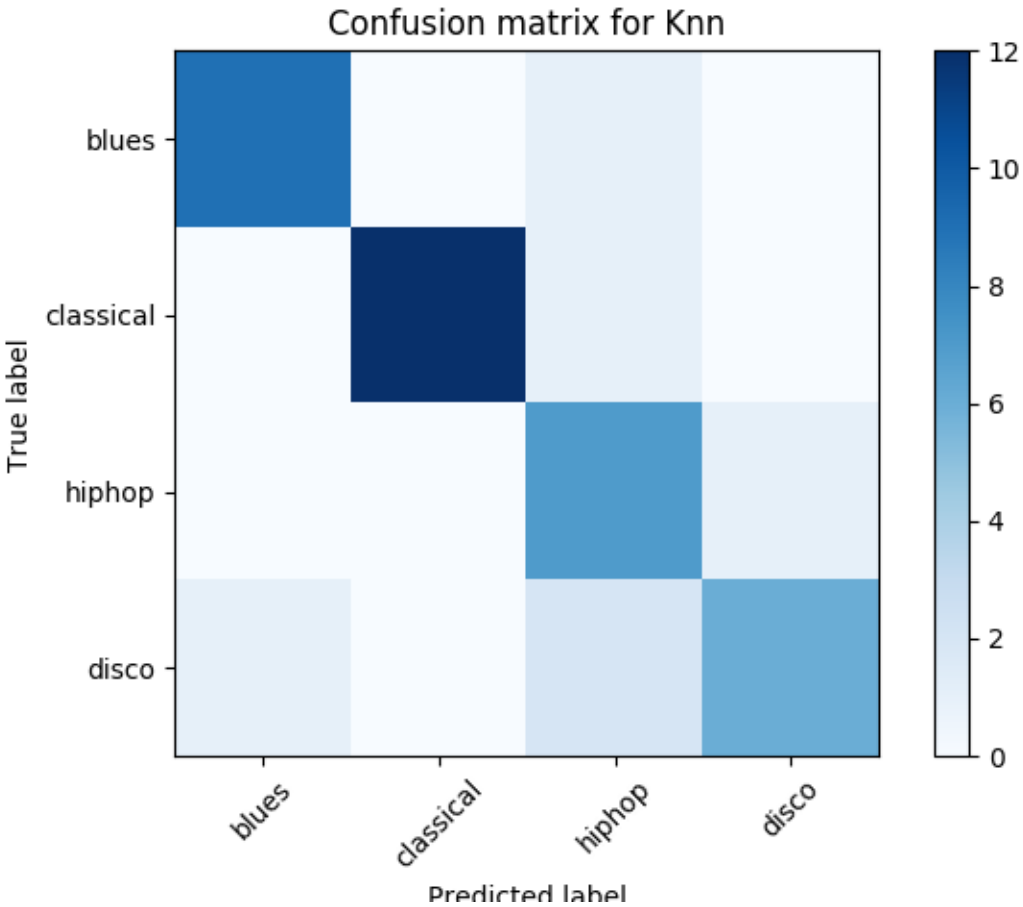

Fig 8.6 MFCC-KNN pair

- MFCC-CNN Confusion Matrix:

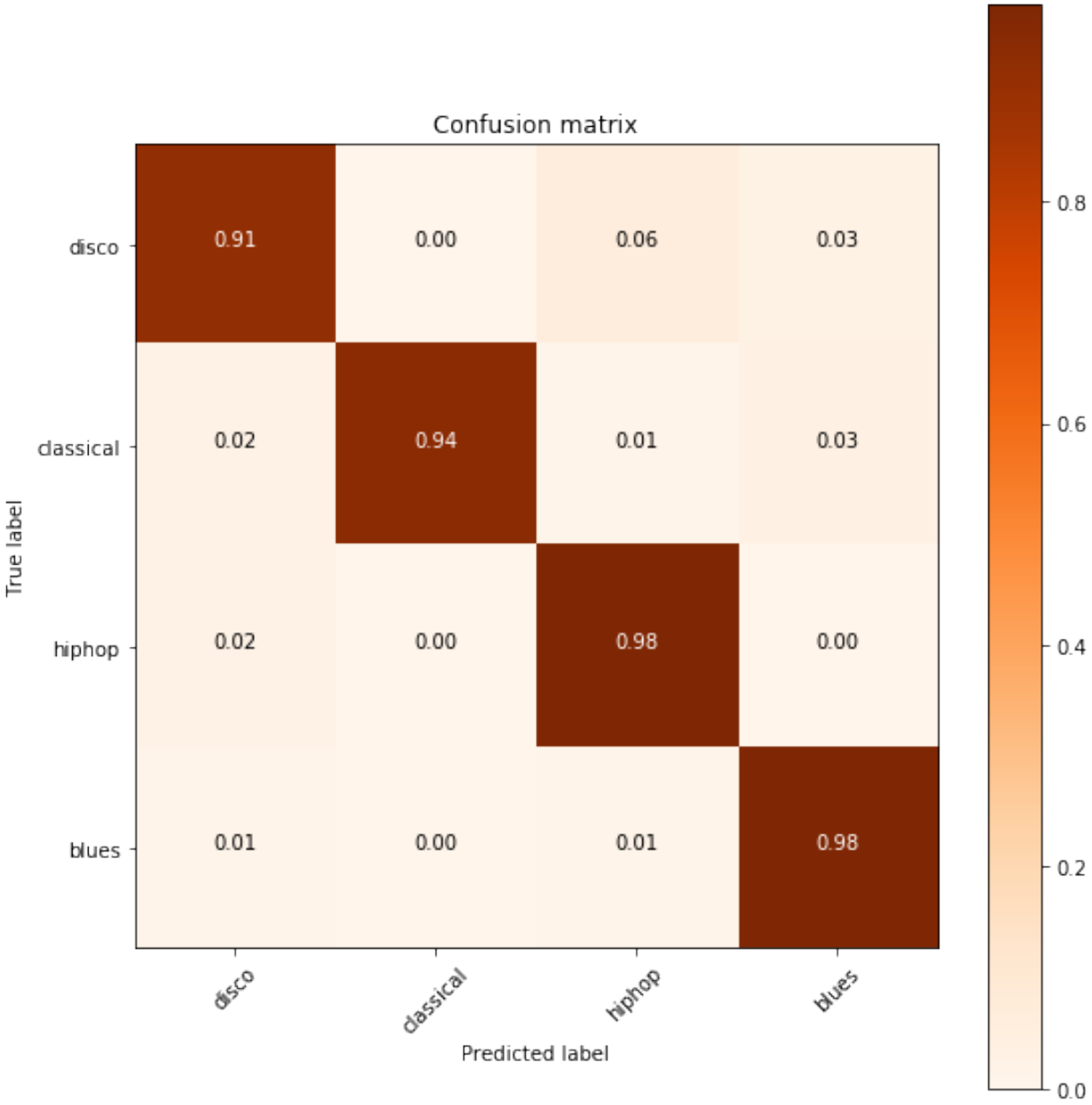

Fig 8.7 MFCC-CNN confusion matrix

- CNN Accuracy Plot:

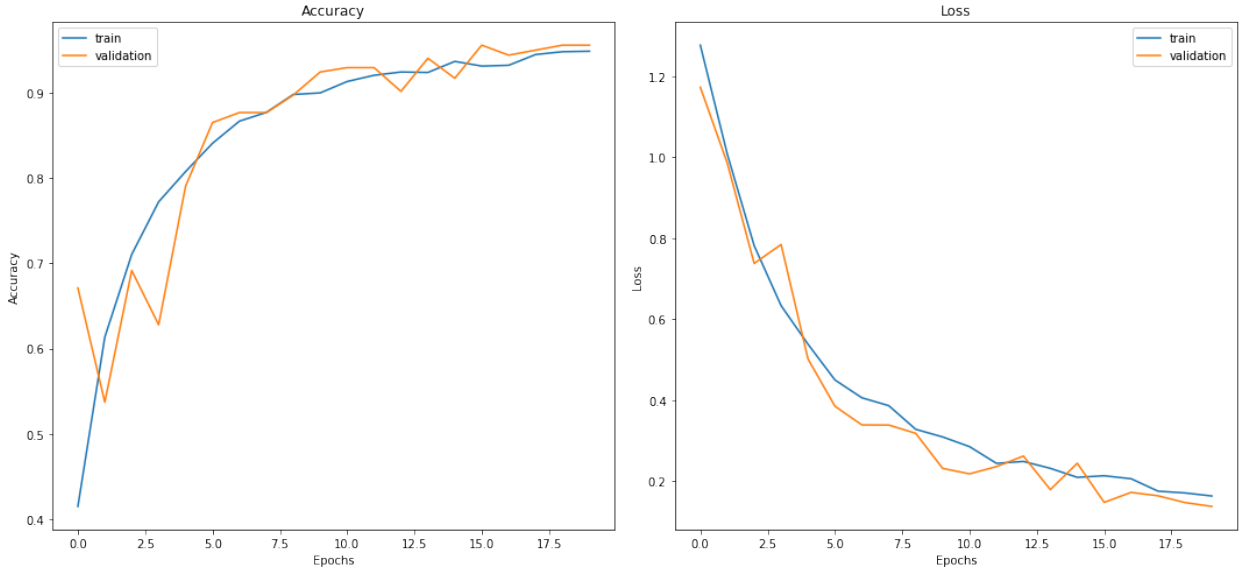

Fig 8.8 CNN accuracy plot

- MFCC Result:

Fig 8.9 MFCC result

- FFT result:

```
logistic accuracy = 0.65
logistic_cm:
[[2 3 0 1]
 [0 7 0 0]
 [1 1 8 2]
 [1 2 3 9]]
knn accuracy = 0.7
knn_cm:
[[ 1  0  1  4]
 [ 0  7  0  0]
 [ 0  0 10  2]
 [ 0  2  3 10]]
svm accuracy = 0.65
svm_cm:
[[5 0 0 1]
 [0 7 0 0]
 [3 0 5 4]
 [1 1 4 9]]
```

Fig 8.10 FFT result

Chapter-9

Conclusions

The outcome of this project is a performance report of various Machine Learning and Deep Learning Algorithms used for Music Genre Classification. The models are trained by extracting features using Fast-Fourier Transforms and Mel-Frequency Cepstral Co-efficients. The model was to train to classify music based on 4 genres – blues, classical, hiphop, disco.

The outcome showed that when FFT is used as the feature extraction method :

- Logistic Regression gives around 67.5% accuracy
- KNN gives around 70% accuracy
- SVM gives around 72.5% accuracy

The outcome showed that when MFCC is used as the feature extraction method:

- Logistic Regression gives around 85% accuracy
- KNN gives around 77.5% accuracy
- SVM gives around 85% accuracy
- CNN gives around 95% accuracy

As a Deep Learning model, CNN was expected to perform better than the other Machine Learning models. As, MFCC is a modern upgrade over the FFT which has been around for some time, It was expected to perform better. Sure enough, the MFCC-CNN pair gave an accuracy of around 95%.

Chapter-10

Future Work

The project has been demonstrated with a basic UI built using Javascript, Html5, Css3 and Bootstrap. In future, the goal is to make the UI more user-intuitive and responsive.

The proposed approach was also applied on “A Database for Indian Classical Music”. It is a database created by IIT Kanpur. In this context, the proposed approach was tried exactly as it is, on the Classical Music Dataset, to check the working of the classification in terms of raagas instead of genres to see if similar approaches work on different datasets. Features were extracted from the dataset in the same exact way. But, a lot of problems were encountered in the classification phase. When the feature set was plotted, some files in the dataset contained skewed data and therefore was difficult to work with.

Hence, in the future, this can be applied to another dataset.

Chapter-11

References

[1] Sturm, Bob L, “An analysis of the GTZAN Music Genre Dataset”

[2] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, Paul Lamere, “Million Song Dataset”, 12th International Society for Music Information Retrieval Conference (ISMIR 2011)

[3] G. Jawaherlalnehru, S. Jothilakshmi, “Music Genre Classification using Deep Neural Networks”, © 2018 IJSRSET | Volume 4 | Issue 4 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099

[4] Alif Noushad, Albin Paul, Anjana Mukesh, Ebin B Plackal, Mohan T D and Anjali S, “A Study on Different Music Genre Classification Methods”, International Journal of Computer Science and Mobile Applications, Vol.6 Issue. 2, February- 2018, pg. 131-138

[5] Y.M.D. Chathuranga, K.L. Jayaratne,“Automatic Music Genre Classification of Audio Signals with Machine Learning Approaches”, GSTF International Journal on Computing (JoC), Vol. 3 No.2, July 2013

[6] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals”, IEEE Transactions on Speech and Audio Processing, 10(5):293{302, 2002.

[7] https://www.cse.iitk.ac.in/users/tvp/music/ website.

[8] Sox.sourceforge.net. Sox - sound exchange | homepage, 2015.

[9] T. Sainath, A. Carmi, D. Kanevsky, B. Ramabhadran: “Bayesian compressive sensing for phonetic classification,” in *Procidin*g *of IEEE International Conferece Acoustic, Speech, Signal Processing*, 2010.

[10] Alan V. Oppenheim, Ronald W. Schafer, “From Frequency to Quefrency: A History of the Cepstrum”

[11] Chungsoo Lim Mokpo, Yeon-Woo Lee, and Joon-Hyuk Chang, "New Techniques for Improving the practicality of a SVM-Based Speech/Music Classifier," IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1657-1660, 2012.

[12] Hongchen Jiang, JunmeiBai, Shuwu Zhang, and Bo Xu, "SVM-Based Audio Scene Classification," IEEE International Conference Natural Language Processing and Knowledge Engineering, Wuhan, China, pp. 131-136, October 2005.

[13] Lim and Chang, "Enhancing Support Vector Machine-Based Speech/Music Classification using Conditional Maximum a Posteriori Criterion," Signal Processing, IET, vol. 6, no. 4, pp. 335-340, 2012.

[14] Md. Al Mehedi Hasan and Shamim Ahmad. predSucc-Site: Lysine Succinylation Sites Prediction in Proteins by using Support Vector Machine and Resolving Data Imbalance Issue. International Journal of Computer Applications 182(15):8-13, September 2018.

[15] https://en.wikipedia.org/wiki/Convolutional_neural_network website.

[16] https://www.cse.iitk.ac.in/users/tvp/music/ website