A Project Report

on

MUSIC GENRE CLASSIFICATION USING CNN

Submitted in the Partial Fulfillment of the Requirements
for the award of

Bachelor of Technology
in
Electronics & Communication Engineering

By

Divas Gupta(20185116)
Priyanka Soni(20185053)
Dipesh Sharma Poudel(20185105)

Under the guidance of

Dr. Vinay Kumar Srivastava
Professor

Department of Electronics & Communication Engineering
Motilal Nehru National Institute of Technology Allahabad
Allahabad – INDIA

Department of Electronics & Communication Engineering
Motilal Nehru National Institute of Technology Allahabad
Allahabad – INDIA

CERTIFICATE

This is to certify that the work contained in the thesis titled "**Music genre classification using CNN"**, submitted by **Divas Gupta**, **Priyanka Soni** and **Dipesh Sharma Poudel** in the partial fulfillment of the requirement for the award of Bachelor of Technology in Electronics and Communication Engineering to the Electronics and Communication Engineering Department, Motilal Nehru National Institute of Technology, Allahabad, is a bonafide work of the students carried out under my supervision.

Date: 02 May 2022
Place: Prayagraj

Dr. Vinay Kumar Srivastava
Professor
ECE Department
MNNIT, Allahabad

<Similarity Index Certificate from IPR Cell>

Acknowledgement

We take this opportunity to express our deep sense of gratitude and hearth felt thanks to our project supervisor, **Dr. Vinay Kumar Srivastava**, Department of Electronics & Communication Engineering, Motilal Nehru National Institute of Technology, Allahabad for his constant guidance and insightful comments during the course of the work. We shall always cherish our association with him for his constant encouragement and freedom to thought and action rendered to us throughout the work.

We are also thankful to our colleagues and friends for their constant support. Finally, we deem it a great pleasure to thank one and all that helped us directly or indirectly in carrying out this work.

Date: 02 May 2022

Place:Prayagraj

Divas Gupta(20185116)

Priyanka Soni(20185053)

Dipesh Sharma Poudel(20185105)

Abstract

This study compares machine learning algorithms to their ability to automatically split song quotes into their genres. First, a review of existing strategies and methods is carried out, both in terms of feature engineering and the provided algorithms. Creating two different data sets from an online online music archive, using sampling and over-sampling to classify classes. Fifty-four features are extracted manually for each sample, using the Librosa and Sound Sound Analysis Libraries. Then two sets of analyzes were performed; initial comparisons, individual features, average values of each type; the second compares each element with the value of f (points calculated using the Scikit-learn's Select K Best function). Four stages were developed - a neural network, a support-vector machine, a random forest and a gradient magnification machine - and training and testing were performed individually, creating each set of two different data sets. The results show that the vector support system is the most suitable algorithm for the job, with Hip-Hop music being the most precisely divided genre. Finally, an evaluation of the entire process is performed, which includes an examination of the database selection, feature selection, and various aspects of project management.

Table of Contents

List of Figures

Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolution Neural Network
DBM	Deep Boltzmann Machine
EM	Expectation Maximization
LSTM	Long Short Term Memory
RNN	Recurrent Neural Network

Chapter 1

Introduction

1.1 Introduction

As per data 24,000 songs get uploaded every day on different platform some major platforms are Spotify, Apple, Google Music, Napster, Dezzer.
so its impossible for us to remember the genre of music with name of music so its very important to classify the music for user easy.
in our project we have taken 10 major genre for classification they are :- **blues, classical, country, disco, hiphop, jazz, reggae, rock, metal, and pop.**
Each genre contain 100 music and each music is 30 sec long.
Our dataset has two folder namely:-

1) **Genres original :-** A collection of 10 genres with 100 audio files each, all having a length of 30 seconds.
2) **Images original :-** A visual representation for each audio file. One way to classify data is through neural networks. Because NNs (like CNN, what we will be using today) usually take in some sort of image representation, the audio files were converted to Mel Spectrograms to make this possible.

Why genre classification:-

Great utility to musical information retrieval systems.Genre is intrinsically built on the similarities between pieces of the same genre and differences between pieces of different genres.An automated genre recognition system would make it possible to classify and search large electronic music libraries.In order to generate better recommendation, user generated rating & reviews, genre or books metadata is used.to give user a user friendly experienced while searching songs. This will also help in browsing songs, not only this our system will help in marketing of song as well. So we are doing genre classification to give awesome user friendly to music lover/listener.

1.2 Motivation

As per data on a single day 60,000 songs streamed in a day. This is really huge almost one songs per second , many people use to be sad and they want to feel happy but they don't know the name of song so they can listen song from classification of happy.

Our project aim to give a systematic approach to store the music data as music data is too vast, we aim to help human sociological and psychological nature, our project aim to save time while selecting songs our project will help to quickly select songs irrespective of occasion, we want to diversify the music system and to help the publisher to define the type of music, Detect the type of music our project can be used as it will give unbiased result as it don't have human intervention. Our main aim motivation is to make music a best friend of music users.

Chapter: 2
Literature Survey

2.1 Introduction to ML

Machine Learning (ML) has been developed from Artificial Intelligence, a field of computer science. Machine Learning (ML) is a multi-disciplinary, mathematical compilation and computer science algorithms widely used in predictable analysis and classification. In recent decades, the proliferation of Artificial intelligence (AI) has become a broad and exciting field in computer science as technology prepares machines for human performance, and aims to train computers to solve real-world problems with a high level of success. As we see the growth of science and technological advances AI systems are now able to learn and improve by using prior knowledge without explicit help code when exposed to new data. Ultimately it leads to machine learning technology (ML) that uses learning algorithms to read from available data. Machine Learning uses data mining techniques to extract information from large-sized databases. ML Methods and Data Mining scan data from end to end to detect hidden patterns within the database. Machine learning and data mining algorithms are invested in various fields such as Computer networking, tourism and tourism industry, financial forecasting, telecommunications industry and power forecasting and more.

2.2 Methods used in Machine Learning

Over the years a large number of ML algorithms have been introduced. Only some of them can solve the problem so they replaced the other one. There are mostly used 3 types of ML algorithms for example supervised learning, unsupervised and reinforcement learning.

2.2.1 Supervised Learning

Contains a given set of input variables (training data) pre-labeled and targeted data. It uses a variable input to generate a map function to map the required output to the corresponding input. The parameter adjustment process continues until the system obtains the appropriate level of accuracy regarding the teaching data.
In other words Supervised learning is a machine learning method that maps out output to the desired inputs based on a pair of output inputs. Considering the work from the training data labeled which includes a set of training examples.

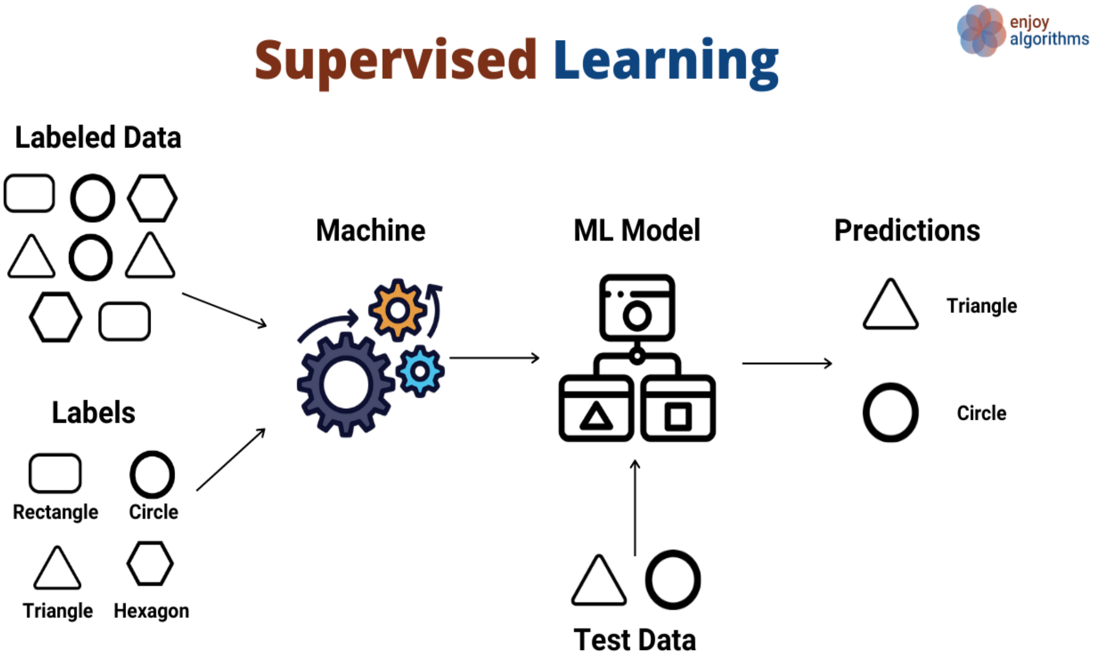


Fig 2.1: Flow chart of supervised learning

The supervised learning algorithm analyzes data given in the form of training and generates activity which we desired in the form of a function, which can be used to map new examples. The right mapping will allow the algorithm to accurately determine class labels in very new situations which are never seen before. This requires a learning algorithm to perform normally from training data to situations that do not appear in a "rational" way.

2.2.2 Unsupervised Learning

In this algorithm we only have training data rather than outcome data. That input data does not have a previous label. It is used for classifiers by identifying existing patterns or collections in input databases. In other words Unsupervised learning is a form of machine learning that looks at patterns that have never been seen in a data set that does not have pre-existing labels and has little human supervision. In contrast to supervised reading that often uses human-written data, Unsupervised Learning, also known as self organization, allows for modeling opportunities for overcrowding. It forms one of the three main stages of machine learning, as well as supervised and reinforced reading.

The two main methods used in Unsupervised learning are the principal component analysis and cluster(group) analysis.

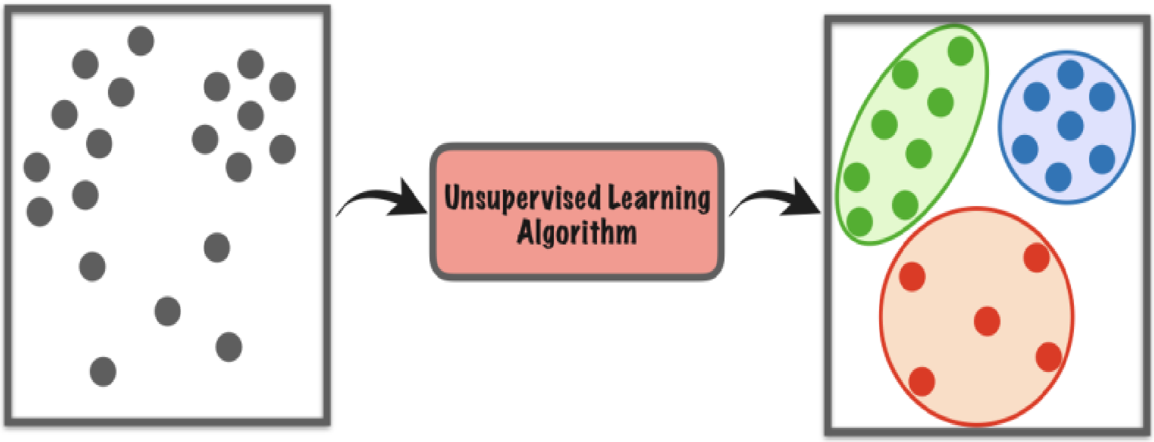


Fig 2.2 : Unsupervised learning

2.2.3 Reinforcement Learning

Using this algorithm machine is trained to map the action to a specific decision which is why the prize or response Symbols are generated. The machine is trained to detect the most rewarding actions by rewards and punishment using previous sensations.

It is a machine learning area about how software agents should take steps in place to develop a vision for the accumulated reward. Reinforcement learning is one of the three basic methods of machine learning, next to supervised reading and non-supervised learning.

Reinforced Learning differs from supervised learning that do not requires paired output/inputs, and require less appropriate steps that need to be clearly adjusted.

Nature is often referred to in the Markov (MDP) decision-making process, because many reinforcement learning algorithms in this context use flexible editing.

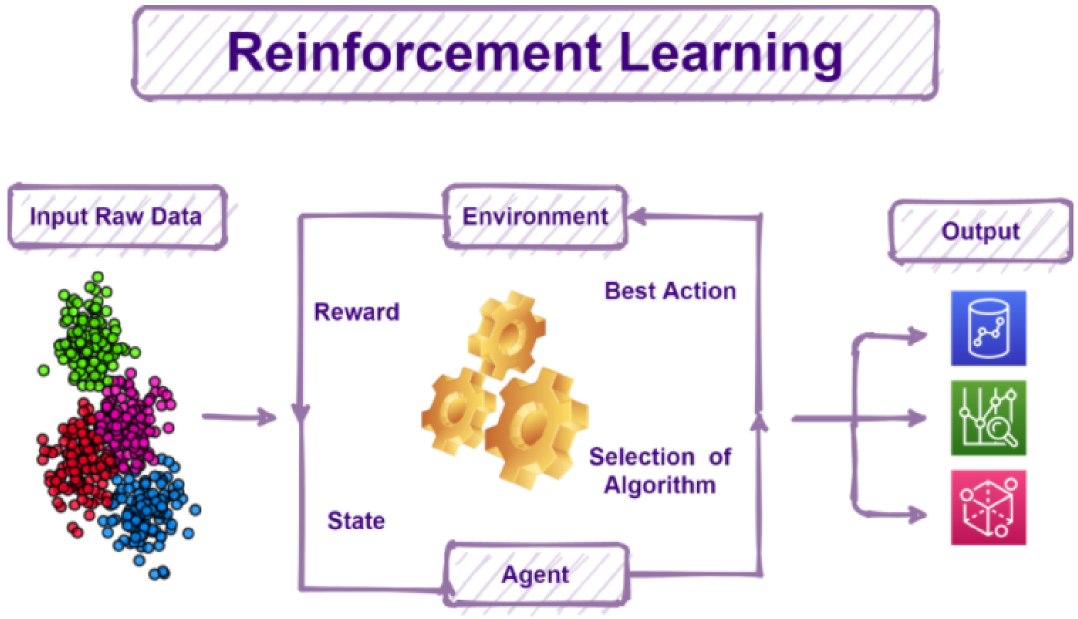


Fig 2.3 Pipeline of Reinforcement Learning

2.3 Algorithm of Machine Learning

2.3.1 Regression Algorithm

In Regression algorithms predictions are modeled by establishing the relationship between variables using error rate. Continuous value is predicted by the Regression strategy. Variables can be value, temperature. Regression algorithms are a type of Supervised algorithms. If we talk about features of this algorithm is that is pays attention on the relationship between the target output and the input features to predict the amount of new data. Algorithms based of regression produces output values on the basis of input feature of the data which is given to the system. Algorithm builds a model on training data features and uses that trained model to predict the amount of new data.

Popular areas of regression algorithms are as follows:

- Linear, Quadratic Regression algorithm
- Normal Decline of Small Squares
- Multivariate Adaptive Regression Splines
- Logistic Regression
- Moderately smooth scatter structure
- Step-by-Step Regression

2.3.2 Decision Trees

Decision trees are most of the used in classification problems like logistic regression. They split attributes in two or more groups by sorting them on the basis of their values. Each tree consists of nodes and branches. Attributes of the clusters are represented by each node and branch represents its value. Pre-pruning and post-pruning are some techniques to improve their accuracy. The most well known algorithms using decision tree are:

- Iterative Dichotomized 3
- Chi squared Automatic Interaction Detection
- C5.0and C4.5 (different versions of a powerful approach)
- Decision Stump
- Classification and Regression Tree
- Conditional Decision Trees

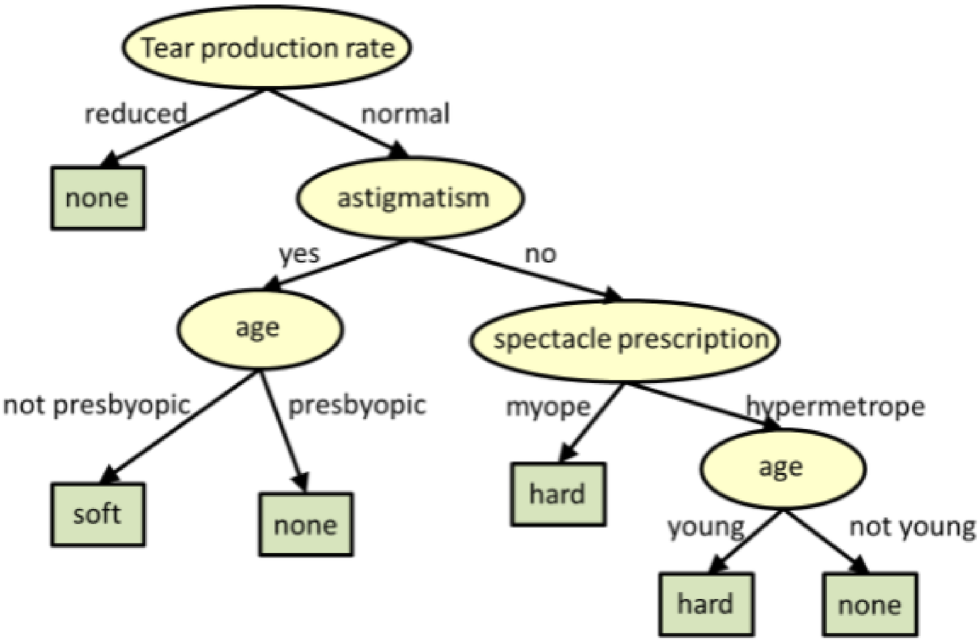


Fig 2.4 Example of Decision Tree

2.3.3 Bayesian Algorithms

Machine Learning is a variety of Computer Science fields like math and algorithm. The statistics control and measure uncertainty and are represented by Bayesian algorithms based on the theory of probability and Bayes theory.

The most common Bayesian algorithms are:

- Bayesian Belief Network (BBN)
- Gaussian Naive Bayes
- Naive Bayes

2.3.4 Clustering Algorithms

This algorithm divides objects into different types of collections. Divides objects into groups where each subset sets share a certain similarity. It is an unsupervised learning method and its methods are categorized as a domain or network clustering and partition clustering. The most popular clustering algorithms are that are used are listed below:

- K Means
- Expectation Maximization (EM)
- K Medians
- Hierarchical-Clustering

2.3.5 Deep Learning Algorithms

Deep Learning techniques are the further advancements made in Artificial Neural Networks. They are more complex neural networks containing lot of hidden layers are large in size. The widely used algorithms for deep learning are:

- Deep Belief Networks
- Deep Boltzmann Machine (DBM)
- Convolution Neural Network (CNN)
- Recurrent Neural Network (RNN)

2.4 Applications of ML

Many industries that work with large amounts of data have recognized the importance of machine learning technology. By obtaining information from this data usually in real time, organizations are able to perform more efficiently or earn more profit than competitors. Other industries that use machine learning are:

Military Applications:

- Inimical energy monitoring
- Monitor friendly forces and equipment
- Observation of military theater or battlefield
- War damage assessment

- Detection of nuclear, biological, and chemical attacks

Nature and environmental Applications:

- Microclimates
- Forest fire detection
- Flood detection
- Agricultural accuracy

Health related Applications:

- Remote monitoring of life data
- Monitor and monitor doctors and patients within the hospital
- Drug administration
- Adult assistance

Home Applications:

- Home automation
- A place with tools
- Automatic meter reading

2.5 ML Shortcomings:

- Each small application requires special training.

- Requires a large amount of training data, organized and structured: Generally required large dataset for more accuracy of model.

- Learning should be monitored regularly, Training data should be marked.

- Requires long time offline / bulk training : For training of a model it takes more memory and time to train model due to large dataset.

- Do not learn increasingly or in an interactive manner , or in groups, in real time.

- Low transfer learning capacity, module usability, and integration.

- Systems and training models are opaque, making it very difficult to find bugs.

- Performance cannot be processed or verified at the ‘long tail’.

Chapter 3:

Methodology and Architecture

3.1 Data Sources

The GTZAN Database, detailed in Tzanetakis (2002), is a well-known source
in the field of MIR. Contains 10 groups of 100 30-second song quotes,
each group representing a single genre, a total of 1000 audio files in total. I
The database will be sorted into folders - one for each type - making it straightforward
roaming, and the nature of the database balance makes it attractive
machine learning projects.
However, despite these attractive features, GTZAN has many drawbacks.
First, with only about 100 items per class, it is a relatively small database, in context
in other machine learning projects. Deep learning is often regarded as thriving
on a large amount of data (Ng, 2015), and having a few training examples means that
class dividers have little knowledge on which to build their models. Therefore
it makes sense to see if a large database will be available.
In addition, Sturm (2013) identified a number of databases, including
repetition of songs (i.e. different quotes from the same song that seem different
tracks in the database) and the non-labeling of tracks (i.e. tracks labeled as
wrong type).
The second database to be considered was the Million Song Dataset (Bertin-Mahieux et al.,
2011). As the name implies, MSD contains a million songs, which are distributed
over a variety of species and subspecies. A million data points is a big data set,
12 Chapter 3. How
and there are many good machine learning projects made with very few uses
examples of training.
However, MSD differs from other data sets such as GTZAN in importance
method: while GTZAN is able to provide access to the audio files of the database itself
(due to the ambiguity of the songs), MSD is collaborating with current celebrities
music, meaning that access to audio files cannot be granted, for reasons of
copyright. A million audio file databases will also take a few terabytes
of data, making it difficult to download and store for most researchers. Instead,
the database contains pre-released features and metadata, such dances as well
thunder, of each song. Although these may be interesting to analyze elsewhere
context, it seems necessary to be able to access audio files
themselves, to allow the release and analysis of similar audio features
the timbre and pitch-content features described above. So, another database became
required.

3.2 Feature Extraction

The database now provides 1000 30-second audio clips, all labeled as one of the 10 possible genres and presented as .au files. For each piece, we made samples a a 2-second window in four random areas, thus enlarging our data into 8000 clips for two seconds each.

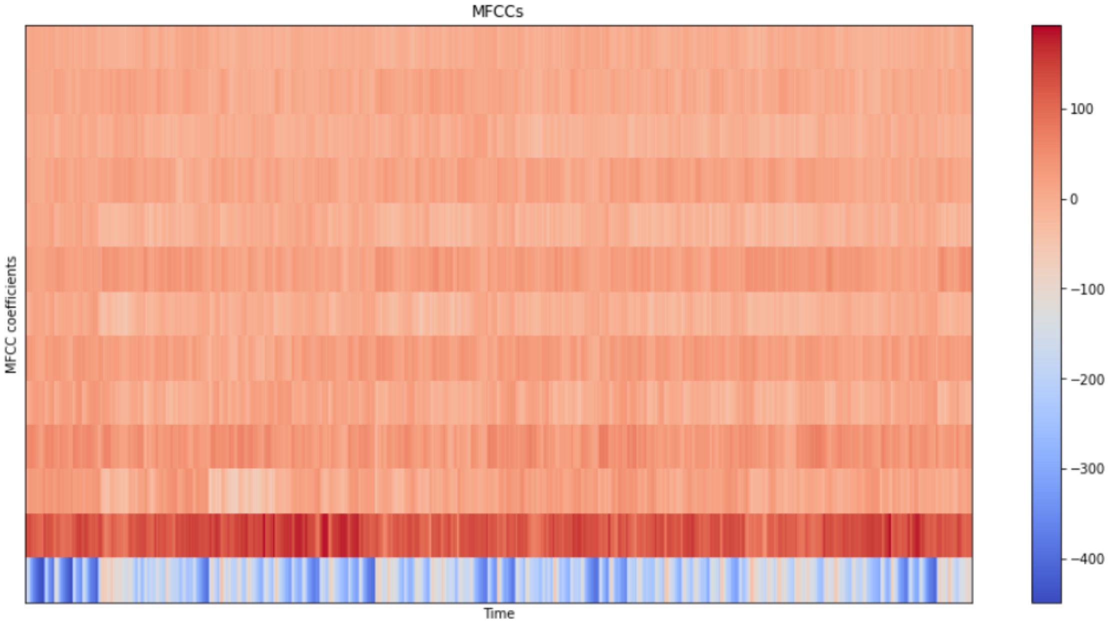
As this data was sampled at 22050 Hz, this leaves us with 44100 raw audio input features. We set boundaries our windows up to two seconds to limit the number of features. We found that 44100 features were a perfect balance between the length of the audio sample and the size of the feature space. So after pre-processing our input is ready (8000,44100), where each element indicates an amplitude peak at a specific time in 44100. We also used 100 samples ofun-augmented data for each of our cross validation and test sets.We also tried to pre-process our data by converting crude audio into mel-spectrograms. By doing this,we have experienced significant performance increases across all models. Mel-spectograms are a commonly used method noisy because they closely represent the way people perceive noise (i.e., log frequency). To convert crude sound in mel-spectogram, one should apply Fourier temporary transitions to smooth sound windows, most usually about 20ms wide. With signal x [n], window w [n], frequency axis ω, and shift m, these are calculated as:

$$\mathrm{STFT}\{x[n]\}(m, \omega) = \sum x[n]\, w[n - m]\, e\ {-j\omega n}$$

These are computed more quickly in practice using sliding DFT algorithms. These are then mapped to the mel scale by transforming the frequencies f by:

$$m = 2595 \log10(1 + f/ 700).$$

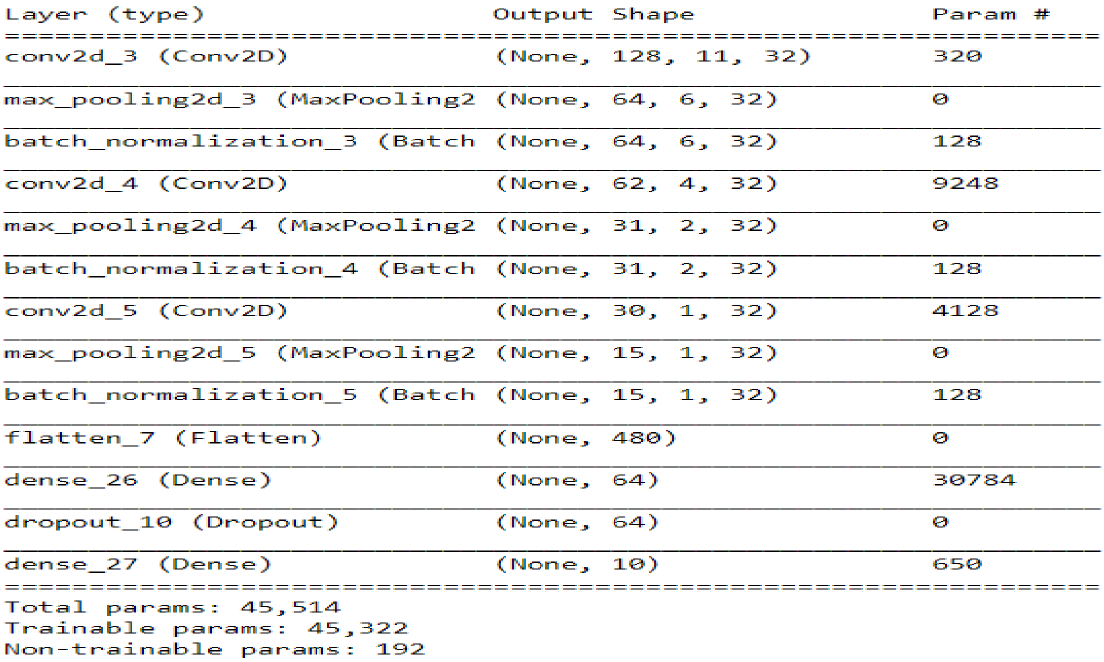
Then we take the discrete cosine transform of the result (common in signal processing) in order to get our final output mel-spectogram. In our case, we used the Librosa library [5] and chose to use 64 mel-bins and a window length of 512 samples with an overlap of 50% between windows. Based on previous academic success with such transformations, we then move to log-scaling using the formula log(X2). The resulting data can be visualized below:


MFCCs
MFCC coefficients
Time
100
0
−100
−200
−300
−400

3.3 Convolution Neural Network

In project, a convolutional neural network (CNN) is used to classify the music genre.The most widely used layer in CNN are: convolution, pooling and fully-connected layers. Also, ReLU(Rectified Liner unit) f(x)=max(0,x) is a type of activation function which is used to add non-linearity. The ReLU works much quicker than conventional f(x) = tanh(x) or f(x) = $(1 + e^{-x})^{-1}$ .Overfitting can be avoided by using dropout layer. The dropout adjusts through output of each hidden neuron to zero with a probability (i.e., 0.5). The neurons which are dropped out cannot contribute to the forward pass and in back propagation.

The main function of CNN is to extract most relevant features from audio. VGG-16 and ResNet are common recommended as image encoders but we have built our own model for this.A brief summary of our CNN model is explained below with a well defined architecture of our model:



```
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d_3 (Conv2D)            (None, 128, 11, 32)       320       
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 64, 6, 32)         0         
_________________________________________________________________
batch_normalization_3 (Batch (None, 64, 6, 32)         128       
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 62, 4, 32)         9248      
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 31, 2, 32)         0         
_________________________________________________________________
batch_normalization_4 (Batch (None, 31, 2, 32)         128       
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 30, 1, 32)         4128      
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 15, 1, 32)         0         
_________________________________________________________________
batch_normalization_5 (Batch (None, 15, 1, 32)         128       
_________________________________________________________________
flatten_7 (Flatten)          (None, 480)               0         
_________________________________________________________________
dense_26 (Dense)             (None, 64)                30784     
_________________________________________________________________
dropout_10 (Dropout)         (None, 64)                0         
_________________________________________________________________
dense_27 (Dense)             (None, 10)                650       
=================================================================
Total params: 45,514
Trainable params: 45,322
Non-trainable params: 192
```

Fig 3.3 CNN architecture

3.4 Artificial Neural Network

We have used a fully connected neural network and, with ReLU functionality and 6 layers, and cross-entropy losses. As input into our model was 1D, when we used mel-spectrograms, we distributed data. Our model is fully connected, which means that each node is connected to all the other nodes in the next layer. For each layer, use it to activate ReLU work at the output of each node, according to the formula:

$$\mathrm{ReLU}(x) = \begin{array}{ll} x & ; x \geq 0 \\ 0 & ; x < 0. \end{array}$$

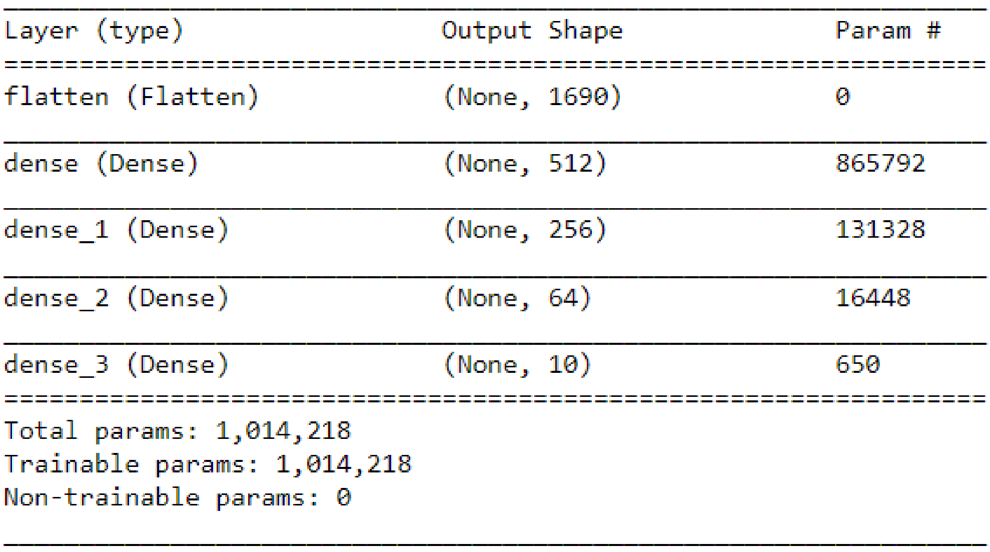
At the end, we construct a probability distribution of the 10 genres by running the outputs through a softmax function:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

To optimize our model, we minimized cross entropy loss:

$$\mathrm{CE}(\theta) = -\sum_{x \in X} y(x) \log \hat{}y(x)$$

We experimented with various regularization techniques, such as dropout layers and L2 regularization. Dropout randomly selects features to drop based off a specified constant, and L2 regularzation adds a penalty term to the loss function



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten (Flatten)            (None, 1690)              0
_________________________________________________________________
dense (Dense)                (None, 512)               865792
_________________________________________________________________
dense_1 (Dense)              (None, 256)               131328
_________________________________________________________________
dense_2 (Dense)              (None, 64)                16448
_________________________________________________________________
dense_3 (Dense)              (None, 10)                650
=================================================================
Total params: 1,014,218
Trainable params: 1,014,218
Non-trainable params: 0

_________________________________________________________________
```

Fig 3.4 ANN Model

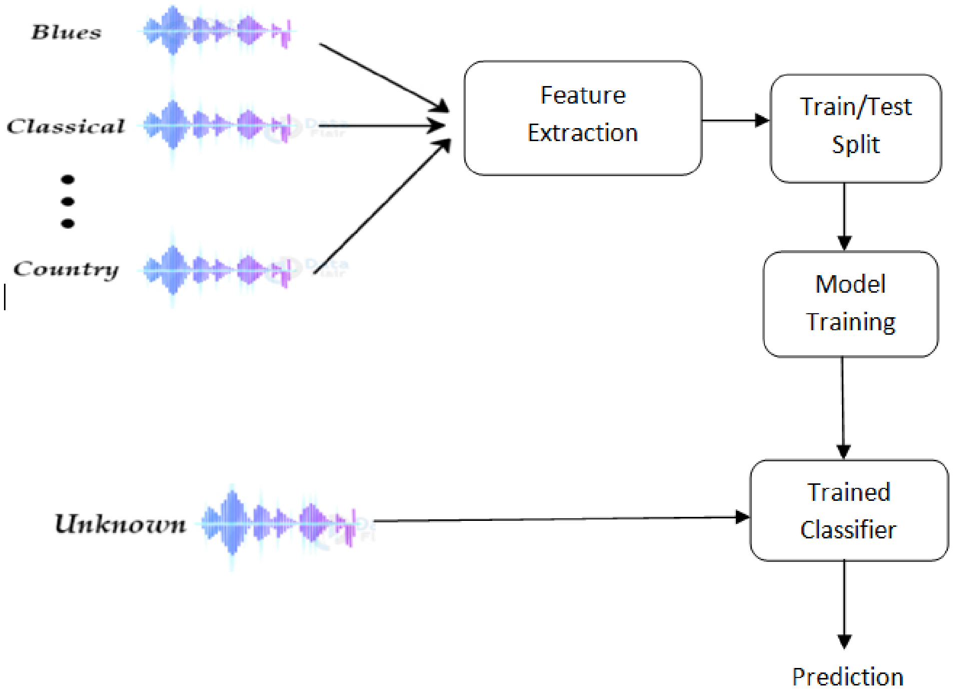
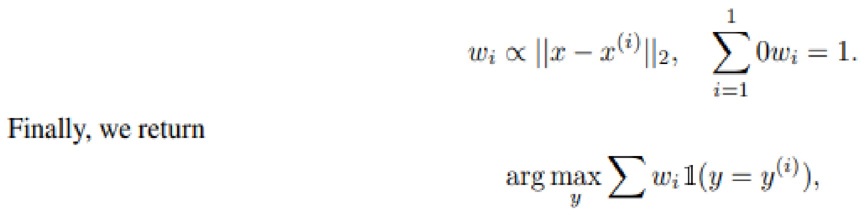


Fig 3.5 Flow chart of the process

3.5 K Nearest Neighbor

After reducing the size using PCA , we used an algorithm for neighbors close to k. Predictions are made on the basis of each case by finding close training examples in our test or validation model that we wish to classify and predicting the label that appeared with greatest frequency among their ground-truth labels. Through trial and error, we found that best accuracy resulted from setting k = 10 and weighting the label of each neighbor by distance.

Explicitly, denoting our data point as x, let x (1), . . . , x(10) be the 10 closest neighbors to x, those which return the largest value on ||x − x (i) ||2, where || · ||2 denotes the Euclidean distance between points. Then, we choose weights wi for each i such that



$$w_i \propto ||x - x^{(i)}||_2, \quad \sum_{i=1}^{1} 0 w_i = 1.$$

Finally, we return

$$\arg\max_y \sum w_i \mathbb{1}(y = y^{(i)}),$$

the label which is most prevalent among x’s 10 nearest neighbors when weighted by distance. This was implemented with scikit-learn

Chapter: 4
Result and Performance Evaluation

4.1 Evaluation Metrics

After building a predictive classification model, to evaluate the performance of the model, i.e. how good the model is in predicting the outcome of new observations of test data that have not been used to train the model.

In other words we need to estimate the model *accuracy* and prediction errors using a new test data set. Because we know the actual outcome of observations in the test data set, the performance of the predictive model can be assessed by comparing the predicted outcome values against the known outcome values.

Various types of evaluation techniques for classification model:-

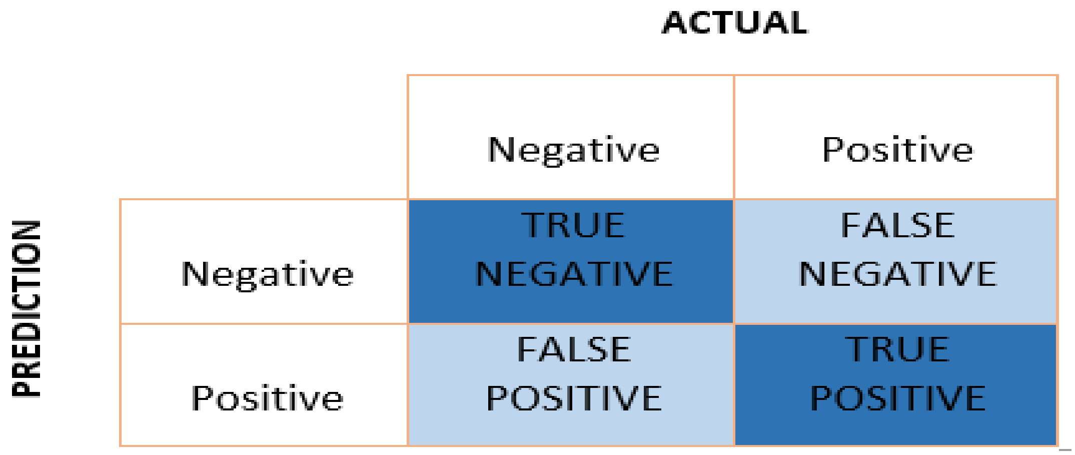
- **Average classification accuracy**: It is representing the proportion of correctly classified observations using a model from a test dataset .
- **Confusion matrix**: It is a 2x2 table showing four parameters, including the number of true positives, true negatives, false negatives and false positives.
- **Precision, Recall and Specificity**: these are three major performance metrics describing a predictive classification model.
- **ROC curve**: It is a graphical summary of the overall performance of the model, showing the proportion of true positives and false positives at all possible values of probability cutoff. The Area Under the Curve summarizes the overall performance of the model.

Introducing Confusion Matrix

Confusion matrix (also known as an error matrix) , is a summarized table/matrix which is used to evaluate the performance of a classification model. The number of correct and incorrect predictions are summarized with count values.

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

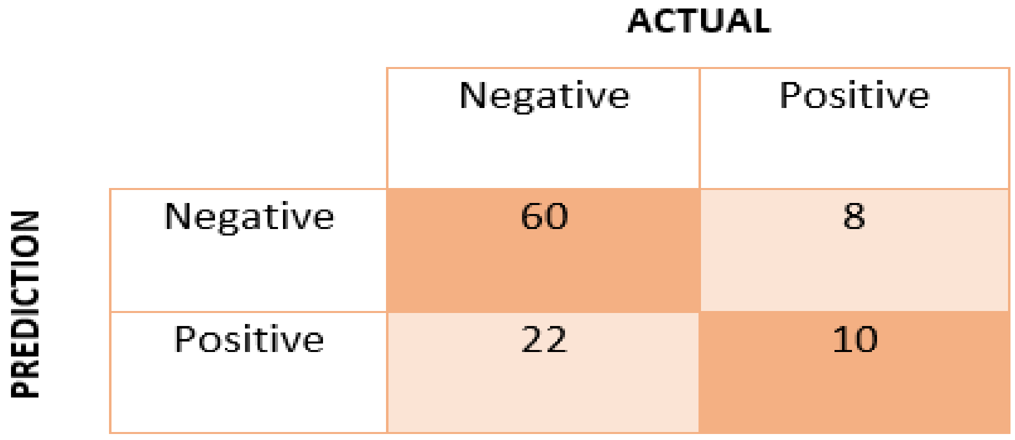
Let's suppose a binary classification problem, so we would have a 2 x 2 matrix as shown below with 4 values:



		ACTUAL	
		Negative	Positive
PREDICTION	Negative	TRUE NEGATIVE	FALSE NEGATIVE
	Positive	FALSE POSITIVE	TRUE POSITIVE

Example: A machine learning model is trained to predict tumor in patients. The test dataset consists of 100 people.

Let suppose the model have confusion matrix like this:



		ACTUAL	
		Negative	Positive
PREDICTION	Negative	60	8
	Positive	22	10

True Positive (TP) — The model correctly predicts the positive class (prediction and actual both are positive). In the above example, **10 people** who have tumors are predicted positively by the model.

True Negative (TN) — The model correctly predicts the negative class (prediction and actual both are negative). In the above example, **60 people** who don't have tumors are predicted negatively by the model.
False Positive (FP) — The model gives the wrong prediction of the negative class (predicted-positive, actual-negative). In the above example, **22 people** are predicted as positive of having a tumor, although they don't have a tumor. FP is also called a **TYPE I** error.
False Negative (FN) — The model wrongly predicts the positive class (predicted-negative, actual-positive). In the above example, **8 people** who have tumors are predicted as negative. FN is also called a **TYPE II** error.

We Can calculate TPR(True Positive Rate), FPR(False Negative Rate), TNR(True Negative Rate), False Negative Rate(FNR) using below formula:

$$TPR = \frac{TP}{Actual\ Positive} = \frac{TP}{TP + FN}$$

$$FNR = \frac{FN}{Actual\ Positive} = \frac{FN}{TP + FN}$$

$$TNR = \frac{TN}{Actual\ Negative} = \frac{TN}{TN + FP}$$

$$FPR = \frac{FP}{Actual\ Negative} = \frac{FP}{TN + FP}$$

Note: The values of TPR and TNR should be high, and FPR and FNR should be as low as possible.

Accuracy:how often is the classifier correct?
(TP+TN)/total

Precision: Out of all the positives predicted, what percentage is truly positive.The precision value lies between 0 and 1.

$$Precision = \frac{TP}{TP + FP}$$

Recall: Out of the total positive, what percentage are predicted positive. It is the same as TPR (true positive rate).

$$Recall = \frac{TP}{TP + FN}$$

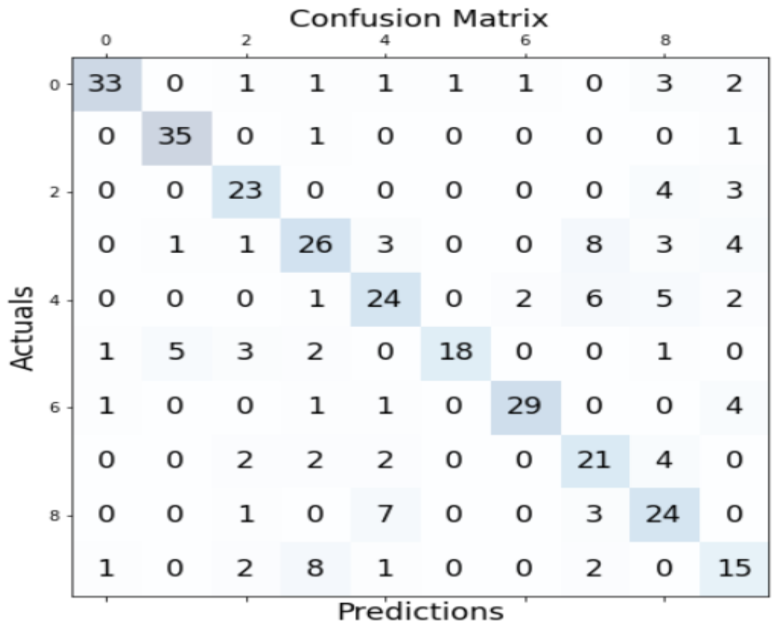
F1 Score: It is the harmonic mean of precision and recall. It takes both false positives and false negatives into account. Therefore, it performs well on an imbalanced dataset.

F1 score gives the same weightage to recall and precision.

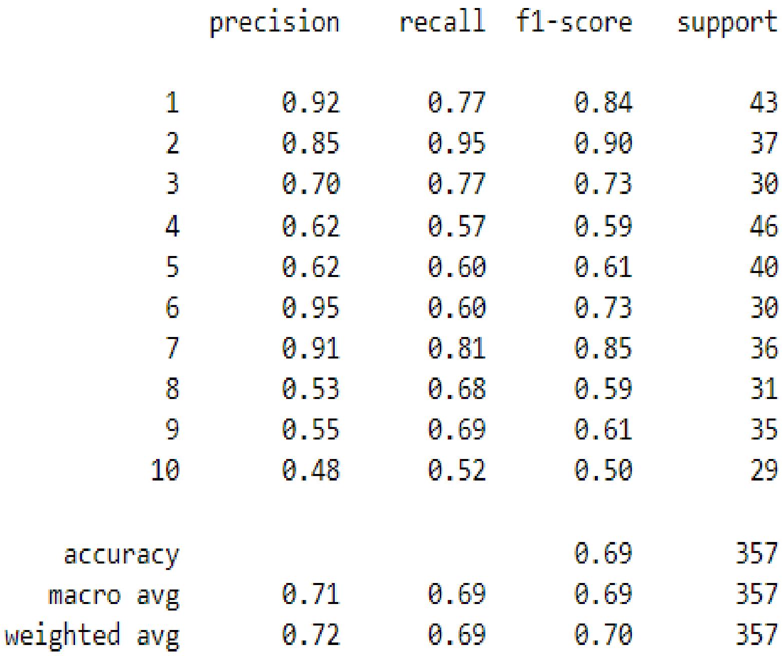
$$F1\ score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2 * (Precision * Recall)}{(Precision + Recall)}$$

4.2 Result

Summary of K Nearest Neighbor ML algorithm:

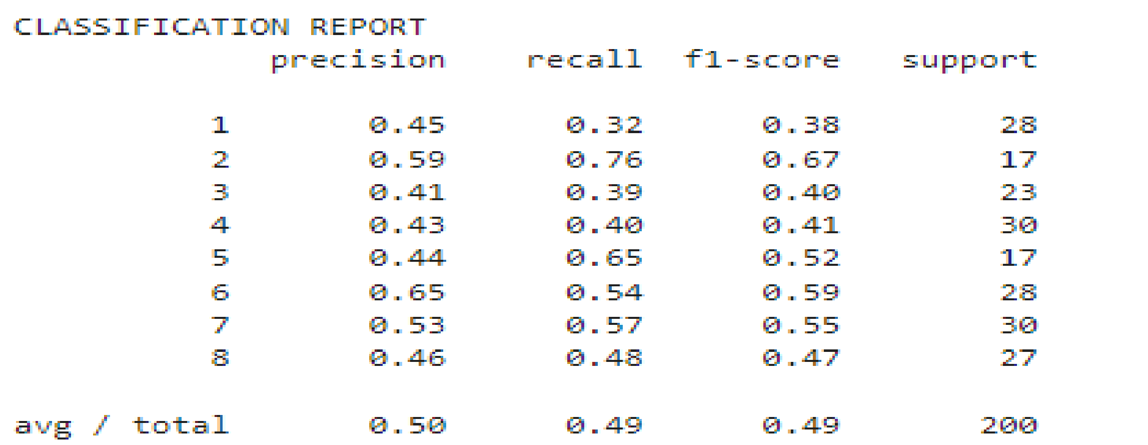


	0		2		4		6		8	
0	33	0	1	1	1	1	1	0	3	2
	0	35	0	1	0	0	0	0	0	1
2	0	0	23	0	0	0	0	0	4	3
	0	1	1	26	3	0	0	8	3	4
4	0	0	0	1	24	0	2	6	5	2
	1	5	3	2	0	18	0	0	1	0
6	1	0	0	1	1	0	29	0	0	4
	0	0	2	2	2	0	0	21	4	0
8	0	0	1	0	7	0	0	3	24	0
	1	0	2	8	1	0	0	2	0	15



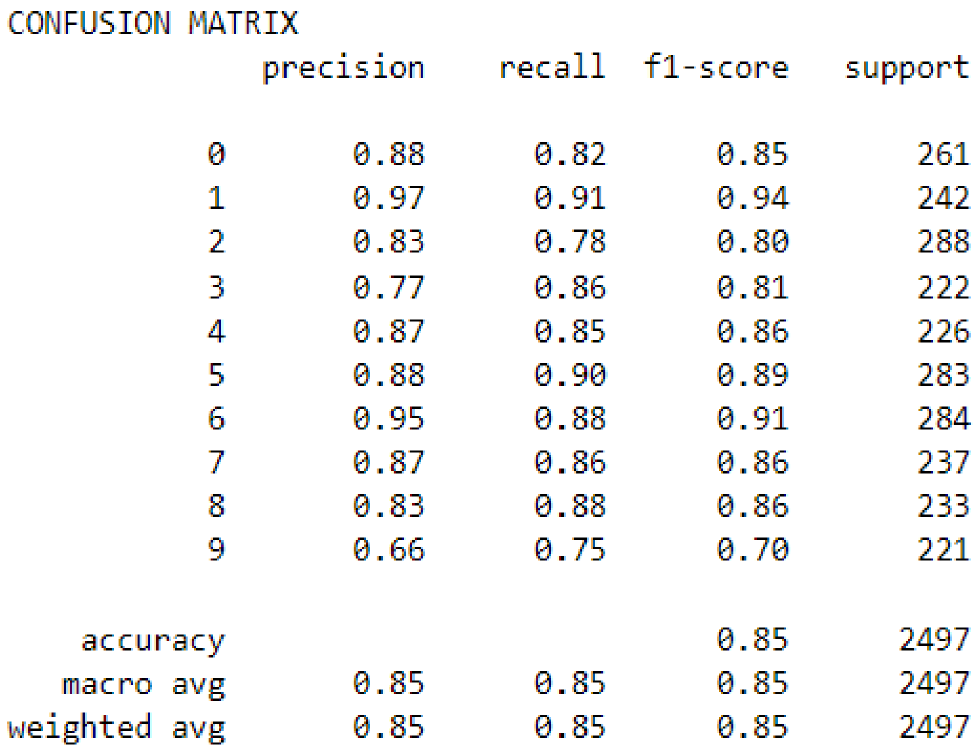
```
              precision    recall  f1-score   support

           1       0.92      0.77      0.84        43
           2       0.85      0.95      0.90        37
           3       0.70      0.77      0.73        30
           4       0.62      0.57      0.59        46
           5       0.62      0.60      0.61        40
           6       0.95      0.60      0.73        30
           7       0.91      0.81      0.85        36
           8       0.53      0.68      0.59        31
           9       0.55      0.69      0.61        35
          10       0.48      0.52      0.50        29

    accuracy                           0.69       357
   macro avg       0.71      0.69      0.69       357
weighted avg       0.72      0.69      0.70       357
```

Summary of Random Forest Classifier ML algorithm:

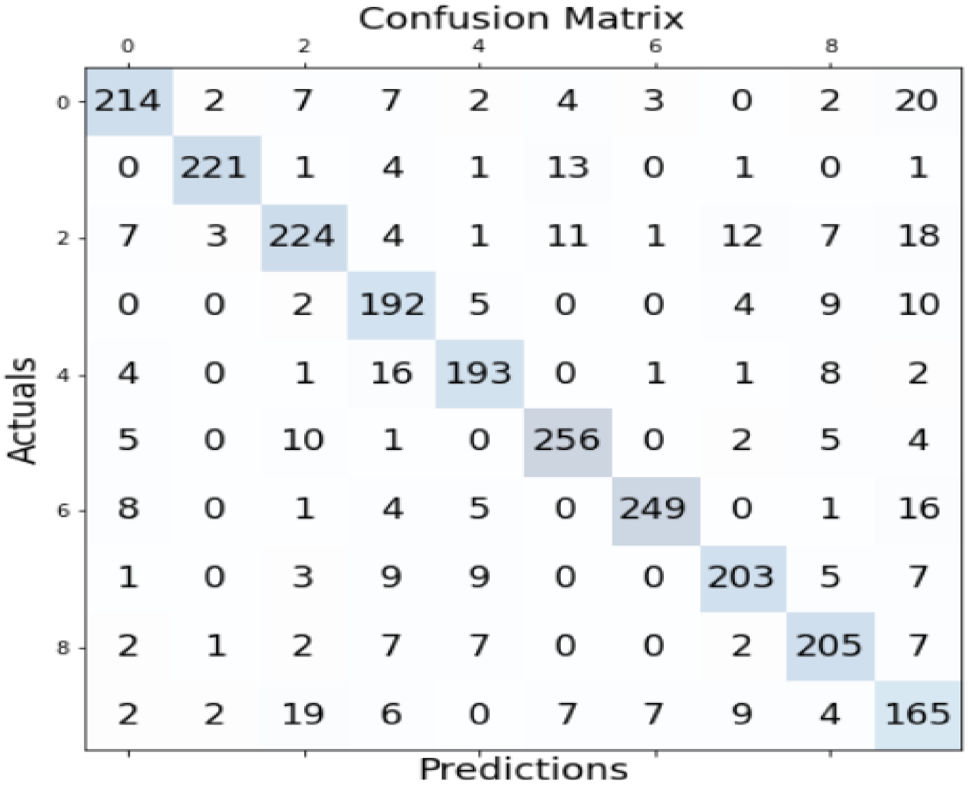


```
CLASSIFICATION REPORT
             precision    recall  f1-score   support

          1       0.45      0.32      0.38        28
          2       0.59      0.76      0.67        17
          3       0.41      0.39      0.40        23
          4       0.43      0.40      0.41        30
          5       0.44      0.65      0.52        17
          6       0.65      0.54      0.59        28
          7       0.53      0.57      0.55        30
          8       0.46      0.48      0.47        27

avg / total       0.50      0.49      0.49       200
```

Summary of CNN Deep Learning algorithm:



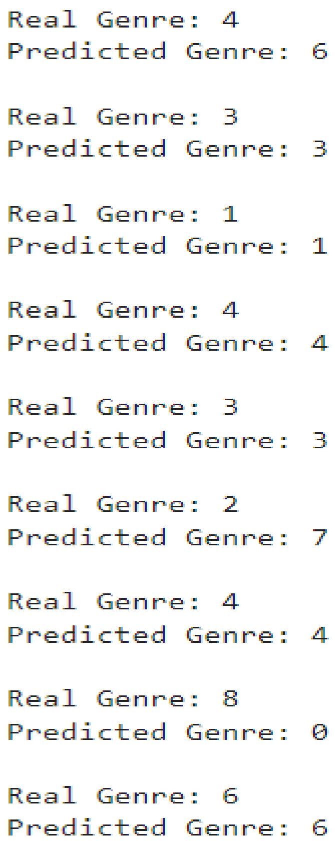
```
CONFUSION MATRIX
              precision    recall  f1-score   support

           0       0.88      0.82      0.85       261
           1       0.97      0.91      0.94       242
           2       0.83      0.78      0.80       288
           3       0.77      0.86      0.81       222
           4       0.87      0.85      0.86       226
           5       0.88      0.90      0.89       283
           6       0.95      0.88      0.91       284
           7       0.87      0.86      0.86       237
           8       0.83      0.88      0.86       233
           9       0.66      0.75      0.70       221

    accuracy                           0.85      2497
   macro avg       0.85      0.85      0.85      2497
weighted avg       0.85      0.85      0.85      2497
```



	0		2		4		6		8	
0	214	2	7	7	2	4	3	0	2	20
	0	221	1	4	1	13	0	1	0	1
2	7	3	224	4	1	11	1	12	7	18
	0	0	2	192	5	0	0	4	9	10
4	4	0	1	16	193	0	1	1	8	2
	5	0	10	1	0	256	0	2	5	4
6	8	0	1	4	5	0	249	0	1	16
	1	0	3	9	9	0	0	203	5	7
8	2	1	2	7	7	0	0	2	205	7
	2	2	19	6	0	7	7	9	4	165

First row represents 261 examples of class ‘0’ 214 are predicted correct.

{0:'blues', 1:'classical', 2:'country', 3:'disco' , 4: 'hiphop', 5:'jazz', 6:'metal', 7:'pop', 8:'reggae', 9:'rock'}



```
Real Genre:  4
Predicted Genre:  6

Real Genre:  3
Predicted Genre:  3

Real Genre:  1
Predicted Genre:  1

Real Genre:  4
Predicted Genre:  4

Real Genre:  3
Predicted Genre:  3

Real Genre:  2
Predicted Genre:  7

Real Genre:  4
Predicted Genre:  4

Real Genre:  8
Predicted Genre:  0

Real Genre:  6
Predicted Genre:  6
```

Fig 4.5 Some of our predictions

Chapter: 5
Conclusion

5.1 Conclusion

In all models, frequency-based mel-spectrograms are used to produce high accuracy results. Although only amplitude provides information on intensity, or how “loud” sound, the frequency distribution provides information in audio content. Additionally, mel-spectrograms are recognizable, and CNN works better with images. CNN
we did our best, as we expected. It took too long to train again, but the increase in accuracy confirms that additional calculation costs. However, we were surprised to see the exact similarities between KNN, SVM, and neural transmission network.

- We see that our CNN struggled most with the rock genre.
- It only managed to correctly classify 70% of rock audio as rock, labeling the others as mainly country or pop.
- Additionally, it incorrectly classified some country, disco and metal as rock music.
- Rock was a challenging genre – a qualitative inspection of rock mel-spectrograms implies that many rock music excerpts lack the easily visible beats that other genres such as hip-hop and disco possess.

5.2 Future Scope of Work

In the future, we hope to try other types of in-depth learning methods, as they have done very well. Given
that this is a time series data, a specific type of RNN model may work well (GRU, LSTM, for example). We also want to know
about the productive features of this project, which include some form of genetic modification (in the same way as production
rival networks repaint images in Van Gogh style, but music in particular). Moreover, we we suspect that we may have opportunities to pass on learning, for example in classifying music by artist or for ten years.

- Can use more genres of music.
- Can use wavelet transform for feature extraction.
- Since, our dataset is a bit small we can combine different dataset to generate a big data which can cover as many variations of music as possible.
- Can extend our work to Speech recognition of different species and also to human’s mood recognition.

References

[1] J.W. Picone, Signal modeling techniques in speech recognition. Proc. IEEE 81, 1215–1247 (1993)

[2] J. Volkmann, S. Stevens, E. Newman, A scale for the measurement of the psychological magnitude pitch. J. Acoust. Soc. Am. 8, 185–190 (1937)

[3] Z. Fang, Z. Guoliang, S. Zhanjiang, Comparison of different implementations of MFCC. J. Comput. Sci. Technol. 16, 582–589 (2000)

[4] G.K.T. Ganchev, N. Fakotakis, Comparative evaluation of various MFCC implementations on the speaker verification task, in Proceedings of International Conference on Speech and Computer (SPECOM) (2005), pp. 191–194

[5] J.S. Mason, X. Zhang, Velocity and acceleration features in speaker recognition, in IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (1991), pp. 3673–3676

[6] Donahue, Jeffrey, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. “Long-term recurrent convolutional networks for visual recognition and description.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2625- 2634. 2015.
.

[7] Denil M, Bazzani L, Larochelle H, et al. Learning where to attend with deep architectures for image tracking[J]. Neural computation, 2012, 24(8): 2151-2184.

[8] Mnih V, Heess N, Graves A. Recurrent models of visual attention[C]//Advances in Neural Information Processing Systems. 2014: 2204-2212.

[9] Vinyals O, Kaiser , Koo T, et al. Grammar as a foreign language[C]//Advances in Neural Information Processing Systems. 2015: 2755-2763.

[10] Hermann K M, Kocisky T, Grefenstette E, et al. Teaching machines to read and comprehend[C]//Advances in Neural Information Processing Systems. 2015: 1684- 1692.