International Journal of Scientific Research and Management (IJSRM)
||Volume||10||Issue||09||Pages||EC-2022-918-927||2022||
Website: www.ijsrm.in ISSN (e): 2321-3418
DOI: 10.18535/ijsrm/v10i9.ec02

Classification of Music genres using Machine Learning

Rachit Manshani, Kanhaiya Suryawanshi, Harshit Mathur

Data Analyst

Abstract
In today's world, Music is a core part of human civilization, it is a form of expression that helps us in conveying our deepest emotions which we are unable to do in simple words. Not only as a form of expression but also a medium to find symphony, peace, and a sense of belongingness in oneself. Since we all usually enjoy different kinds of music genres as per our preferences, wouldn't it be bliss if we had some kind of framework to categorize these melodies into different genres

Keywords : Music Genre Classification, Machine Learning, Support Vector Machine (SVM) algorithm, K-nearest neighbors (KNN) algorithm, Random Forest algorithm, Artificial Neural Network (ANN), Logistic Regression algorithm.

1. Introduction

We In order to further grasp the idea, let's consider a scenario. Suppose, you are listening to a song by Ed Sheeran which is of the Genre - Indie and has the equalizer setting of a general Indie song but the upcoming song in my playlist is by Led Zeppelin, which is of the genre Hard Rock. Now, the experience of listening to the song will get ruined because of the previous equalizer setting of Indie music. In order to prevent the same thing from happening, our algorithm will help in predicting the genre and its appropriate equalizer setting which in turn will enhance the overall experience.[4]

It's important to understand that usually, equalizers are set flat so you can hear the sound as it was originally recorded. However, you can always improve your audio experience by tweaking the settings according to the genres of music you're listening to at the moment and also according to your speaker's capabilities.

Hence, within this research we are going to solve the above stated problem using appropriate pre-processing and Machine learning techniques.[7]

2. Work Done

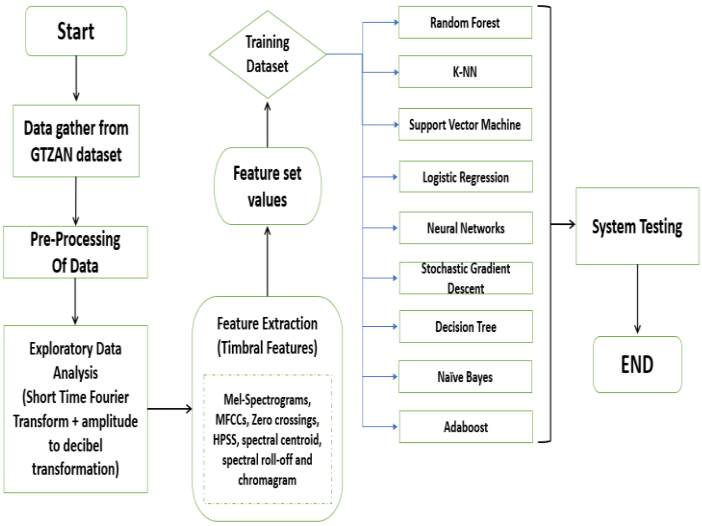
Tzanetakis and Cook (2002) used features such as Mel-Frequency cepstral coefficients (MFCCs), spectral contrast and spectral roll-off for music genre recognition. They proposed these features for direct modeling of music signals and used K-Nearest Neighbors and Gaussian Mixture models for music genre classification.

Lidy and Rauber (2005) discussed the contribution of psycho acoustic features for music genre recognition and much focus was directed on STFT taken on Bark scale.

Nanni et al (2016) used a combination of acoustic and visual features to train SVM and AdaBoost classifiers. Hrishikesh Deshpande, Unjung Nam and Rohit Singh (2001) used Support Vector Machines, Gaussian Mixtures and Nearest neighbors for classification of music into rock, zazz and piano depending on their timbral features.[1]

S.Vishnupriya and K. Meenakshi (2018) used Convolutional Neural Networks to present a music genre classification framework. They used GTZAN dataset and MFCCs for feature extraction. Librosa package was used for feature extraction in python and learning accuracy obtained using MFCC was 76%.

3. Approach



3.1 Determining the dataset

After analyzing several datasets, we decided to proceed with the GTZAN dataset for the sake of this research as we felt that the above-mentioned dataset catered to our needs. There are over 1000 tracks present in the GTZAN dataset. There are 10 genres in it, each containing 100 songs. All the tracks are 30 seconds long, 22050 Hz Mono 16-bit and in .wav format.

The genres are:

1. blues
2. metal
3. classical
4. country
5. disco
6. pop
7. reggae
8. rock
9. hiphop
10. jazz

Some other datasets that we considered before finalizing the GTZAN dataset were the

- ISMIR 2004 dataset (Contains a range of 300 to 350 classical, 100 to 120 electronic,100 to 110 rock/pop, 20 to 30 Jazz/Blues, 40 to 60 Metal/punk and 100 to 140 world songs. The format for all the audio files is .MP3. The dataset is categorized into 3 different folders namely Training, Evaluation and Development, each folder containing 729 files)
- Latin music dataset (This dataset includes around 3000-3500 music songs of over 500-550 different artists dividing into 10 genres)
- African music dataset (This dataset is categorized in three distinct sets: a balanced training set, a balanced evaluation set, and an unbalanced training set)

3.2 Loading and trimming our dataset

Once we have determined the dataset to be used, we then proceed to load the dataset in our workbook.

We used the LaRosa python library for all the music and audio analysis. Librosa provides all the building blocks which are crucial to create music information retrieval systems, many of which we utilized in this research. Librosa also helps to visualize certain audio signals and extraction of features by using various signal processing techniques.

We also trimmed any leading and trailing silence from our audio signals using the Librosa module.

3.3 EDA - Exploratory Data Analysis

Firstmost, we performed a Short Time Fourier Transform (STFT) on our audio file using the Librosa module. STFT is used to represent a signal in a time-frequency domain by computing certain discrete Fourier Transforms over short overlapping windows. We will be keeping the HopLength to 512 and FFT length to 1024.

This was followed by converting the amplitude spectrogram to a dB-scale spectrogram using the amplitude_to_db function of the Librosa module. The reason for doing so is to make the scale logarithmic from a linear scale thus limiting our numerical range. Hence when plotted, the intensity of colors corresponds more closely to what we actually hear.

3.4 Feature Extraction

Spectrograms, MFCC (Mel Frequency Cepstral Coefficients) and Mel Spectrograms are often used for feature extraction of audio files.[13]

Spectrograms: A spectrogram is used to display the strength of a given signal over time at a waveform's various frequencies. Spectrograms are depicted using a two-dimensional graph with colors describing the third variable.

If a time domain signal is divided into shorter segments of equal length, we can create a spectrogram. Then to each segment, FFT (Fast Fourier Transform) will be applied. The plot of the spectrum on each segment is our resulting spectrogram.

MFCC (Mel Frequency Cepstral Coefficients): The MFCCs of any signal are a small set of features describing the overall shape of a spectral envelope. MFCCs are often used in Music Information Retrieval (MIR) to describe timbre.

Conventional approach to develop MFCCs are:

1. Conversion of Hz scale to Mel scale.
2. Calculating the Logarithmic of Mel representation of the audio file
3. Taking logarithmic magnitude and performing Discrete Cosine Transformation

By following this procedure, we can create a MFCC by creating a spectrum over Mel Frequency as opposed to time.

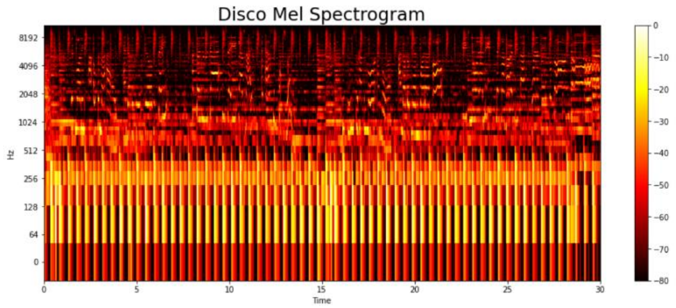
Mel Spectrograms: Mel Spectrograms visualize sounds on the Mel scale as opposed to the frequency domain

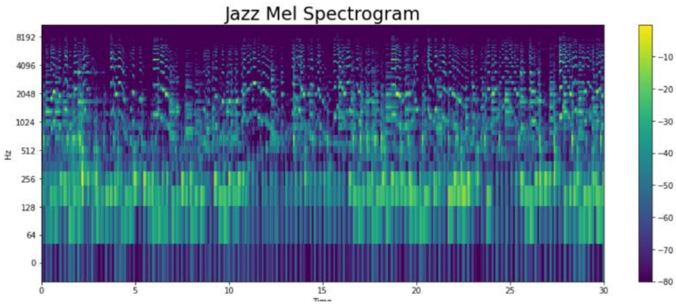
Mel scale is used for relating perceived frequency (pinch of a pure tone) to its measured frequency.

The formula for conversion of frequency to mel scale is:

$$M(f) = 1125 \ln (1 + f/100)$$

Firstmost, we computed and plotted the Disco Mel Spectrogram and Zazz Mel Spectrogram by using Librosa module and its functions particularly librosa.feature.melspectrogram (used to compute a mel-scaled spectrogram. We mapped the frequency onto a mel scale to form the mel spectrogram) and librosa.display.specshow (Used to display a spectrogram or a chromagram).





The next step in the feature extraction of our audio file was to

compute the zero crossing rate of our audio time series by using the Librosa module and its function, librosa.feature.zero_crossing_rate.

The instantaneous point where the sign or polarity of a function (mathematical) shifts, say from positive to negative is known as the Zero crossing.

The rate at which a signal transitions from negative to zero to positive or vice versa is known as the zero-crossing rate (ZCR). ZCR is often used in Music Information Retrieval for identifying percussive sounds.

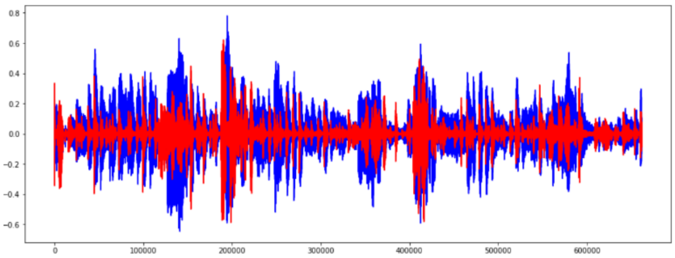
Mathematically ZCR can be represented as:

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

Thereafter, we proceeded with HPSS (Harmonic-Percussive Source Separation) of our audio file by using Librosa.effects.hpss to decompose our audio source into its harmonic and percussive components.

Harmonic sounds are those sounds which we perceive to have a certain pitch. A violin would be a suitable example of a harmonic sound.

Percussive sounds often originate by the collision of two objects like the collision of two castanets. An important feature of percussive sounds is that even though they do not have a definite pitch they have a quite clear localization in time.



After performing HPSS on our audio file, we proceeded to compute the spectral centroid for our audio file using the Librosa module and the function librosa.feature.spectral_centroid by which each frame of the magnitude spectrogram under consideration is normalized and treated as a distribution of the frequency bins. From here the centroid/mean is extracted from each frame.

Spectral Centroid can be defined as the location of the center of mass of the spectrum under question hence spectral centroid is useful for the characterization of the spectrum of the audio file. Mathematically, spectral centroid can be represented as:

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

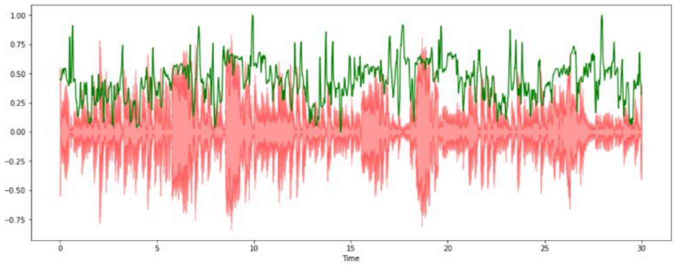
where

- x(n) is the weight frequency
- n being the bin number
- f(n) is the central frequency of the bin

Using the spectral centroid, one can predict the brightness of an audio file and hence it is widely used in measuring the quality of tone of a music file.

Once the spectral centroid has been computed, we decided to proceed with the computation of the roll-off frequency of our audio file via the Librosa module and the function librosa.feature.spectral_rolloff.

The action of a specific filter to roll off frequencies outside a specific range is called a **Spectral Rolloff.**. We can also argue that the spectral roll off point is the fraction of bins where 85% power is at lower frequencies in the power spectrum. This is used for determining the minimum and maximum by initiating the roll percent close to 1 or 0 depending on our needs.

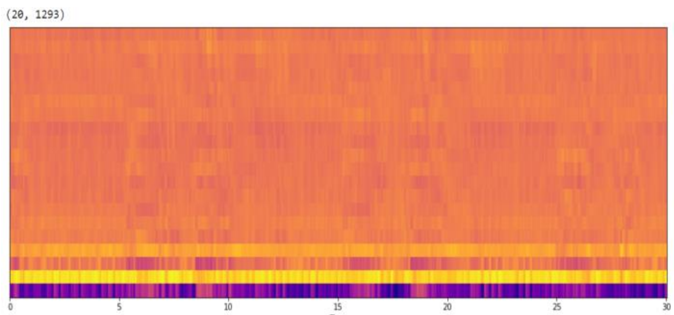


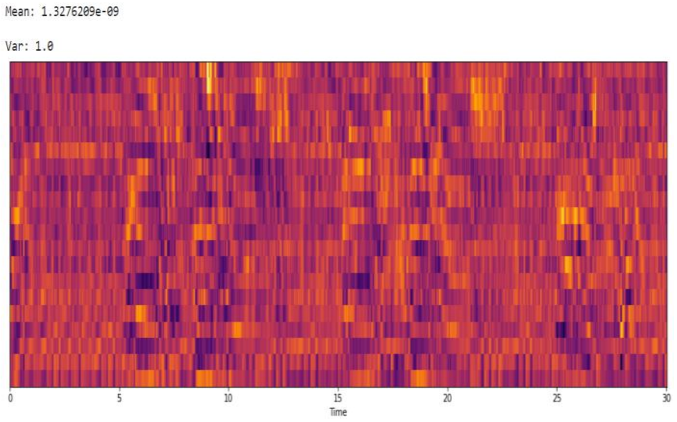
After performing the spectral rolloff, we determined the mean and variance for the wave function with the help of **MFCC**.

Mel Frequency Cepstral Coefficients of any signal are a small set of features describing the overall shape of a spectral envelope. MFCCs are often used in Music Information Retrieval (MIR) to describe timbre.

First, we'll compute a MFCC across an audio signal and determine the shape for our MFCC.

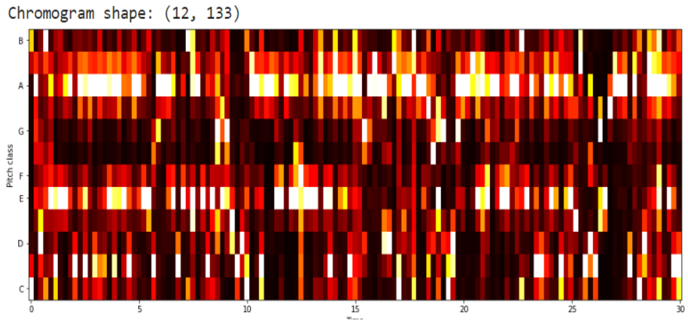
To determine the mean and variance for the MFCC, we used sklearn.preprocessing.scale which is used to standardize a dataset along any axis.





Once we have determined the mfcc shape and its corresponding mean and variance, we proceeded with plotting a chromagram using hop length 5000.[14]

Chromagram is a spectral audio mapped on an octave (12 bins each), in simple words it is a tool for analysis of music files where categorisation of pitch is relevant. We plotted the chromagram by using the librosa module and the function librosa.feature.chroma_stft.



3.5 Scaling

After performing the Data Preprocessing to create appropriate variables such as mean, variance, spectral rolloff, spectral centroid, etc. We need to use PCA for feature scaling. Henceforth we have to use one of the 2 scaling techniques: **Standard Scaler, or the MinMax Scaler.** We have used MinMaxScaler for this purpose, as Standard Scaler transforms features independently to a unit variance and zero centered. The data value range is fixed between 0 and. Whereas. **MinMax Scaler** has the agility to set the minimum and the maximum range of data value. e.g., -1 to +1, -10 to +10. It should be used when it is required to capture small variance in features and also for sparse data where zero value needs to be preserved, which is the case for us. We need the slightest of the variance to be captured for our classification model

4. Principal Component Analysis

It forms the basis of **multivariate data analysis (MVDA)** based on projection of multiple variables onto a single line/plane. The most important use of PCA is to represent a multivariate data as a smaller set of variables (summary indices) for observing clusters, trends and the outliers. This helps in observing the relationships between observations and variables, and among the variables themselves.

5. Model Selection

Machine learning is a branch of AI (artificial intelligence) and computer science that allows a software or an application to become accurate at predicting certain results without being programmed to do so as machine learning enables the use of historical data as input to predict new outcomes.[11]

There is a wide variety of ML algorithms which can be categorized into three main groups:

1. Supervised Learning: Under this category, our machine learns under supervision. Our models are able to predict a certain outcome based on a labeled dataset. Here labeled dataset means a dataset where we already know the target answer.

   We generate a function for mapping our inputs to desired outputs by using independent variables or predictors. We perform this task repeatedly until our model achieves a certain degree of accuracy.

   Depending on the characteristics of the target variable, it can be a discrete target variable task/Classification or a continuous target variable task/Regression.

- Classification Supervised Learning: This approach is used to identify a category of new observations based on the training data. Here, a program learns from the dataset under consideration and then classifies the new observation made into groups or classes. Classes could be like Yes or No, Cow or Buffalo etc.

  The main goal of this algorithm is to identify the category of a given dataset. Some examples of classification algorithms are Logistic Regression, K nearest neighbors and Naive Bayes.

- Regression Supervised Learning: When the output variable is a real/continuous value, regression algorithms are used. In the case of regression algorithms, there is a relation between two or more variables which means that any adjustion in one variable is followed by a change in another variable.

  Regression is mainly used for forecasting, time series modeling, determining market trends etc. Some important types of regression algorithms are Linear regression, decision tree regression etc.

2. Unsupervised Learning: In this algorithm, we do not have the luxury of a target/outcome variable that we can predict or estimate. Hence, in case of unsupervised learning, a model is prepared by deducing certain structures which can be present in the input data. This is usually done to extract a set of general rules. It is done to either reduce redundancy through a mathematical process or to organize data by similarity. Some example problems are clustering, dimensionality reduction etc.

   Some examples of unsupervised algorithms are K-means clustering, Principal component analysis, Hierarchical clustering.

3. Reinforcement Learning: This type of learning works on an action-reward principle. The machine under the influence of this algorithm makes certain decisions and is continuously trained by using trial and error. Then it learns from previous experimentations and utilizes the most suitable pathway to make appropriate decisions. One such example of this learning is the Markov Decision process.

After considering all the algorithms we decided to proceed (i.e., algorithm that has the best accuracy).

1. *Naive' Bayes Algorithm:*

The Naïve Bayes algorithm is composed of two words Naïve and Bayes, it is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. It is called Bayes because it depends on the principle of Bayes' Theorem. Suppose, consider a dataset of weather conditions and respective aimed variable as "**Play**", follow, the data set is converted into frequency tables, form a new table by finding the probabilities of mentioned features and now, calculate the posterior probability using bayes theorem.[2][8]

Formula used: $P(y|X) = \frac{P(X|y)P(y)}{P(X)}$

2. Stochastic Gradient Descent:

This algorithm is iterative which begins from a random point on the function and moves down the slope until it finds the lowest point of the function. The algorithm picks a random point from the data set which is called a mini batch and this decreases computations. It is a balance between advantages of gradient descent and speed of the algorithm.

Objective function : $$\text{for } i \text{ in range } (m): \quad \theta_j = \theta_j - \alpha\left(\widehat{y}^i - y^i\right)x_j^i$$

3. K-Nearest Neighbors (KNN) algorithm:

It is one of the simplest modeling algorithms. It basically works by storing data in different categories by dividing the training data accordingly. Whenever a new data comes, it is being checked with the groups to which set it belongs to and then identification is done.

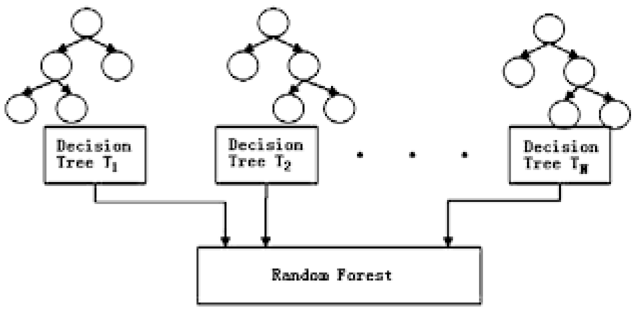
Working of KNN: Select the neighbors then calculate the euclidean distance, further count the data points and assign them in categories accordingly where the number of neighbors are maximum[12]. Model is ready.

4. Random Forest:

It is a popular machine learning algorithm which does both classification and regression problems, uses a concept of combining multiple classifiers to improve data modeling performance (Ensemble learning).

It takes less training time, has high accuracy and maintains that accuracy for large proportions of data.

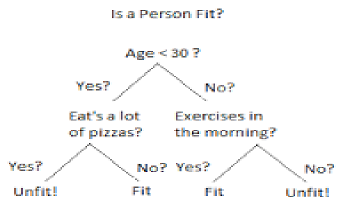
Working: Select K data points and build decision trees on them, then choose number N for the decision trees and again repeat the first two steps. When new data points come, find predictions of each tree and then assign to whoever has maximum count(vote).



4. Decision Tree classification:

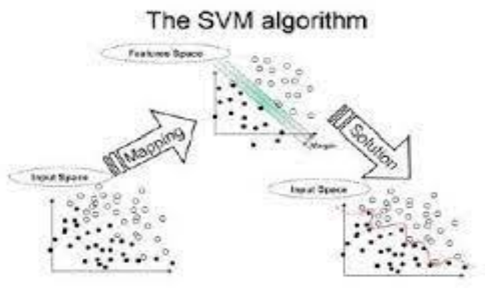
Like random forest it also can be used for both classification and regression but is used for classifications, in a tree structured classifier, internal nodes represent features of the dataset, branches are decision rules and leaf nodes are outcomes. It is a graphical representation of all possible outcomes(solutions) to a problem. It gives a human-like thinking, thus easy to understand. major terms are: leaf node, root node, branch, parent etc.

Example working:



5. Support Vector machine algorithm:

This algorithm can also be used for both classification and regression but is only used for classification. This algorithm tries to create the best decision line or boundary which can differentiate the data into classes so that we can easily put new data into categories easily. It finds the closest point of lines from all the classes; this region is called the Hyperlane. The maximum is called optimal hyperplane.[9]

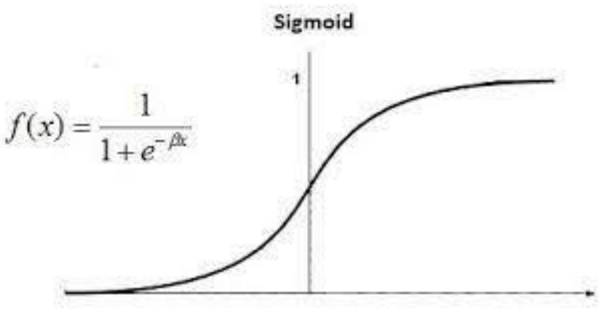


6. Logistic Regression machine algorithm:

It is a very popular technique for supervised learning, used to predict categorical dependent variables using independent variables. The working of this model is very similar to that of Linear regression, the only difference is that linear regression is for regression problems whereas logistic regression is for classification related problems.

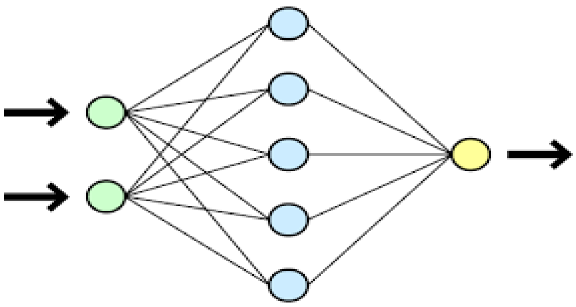
The equation used by the basic logistic model is Ln (p/1-p) =a0+a1*x+a2*x (1) This is called the logistic function.

Principle graph:

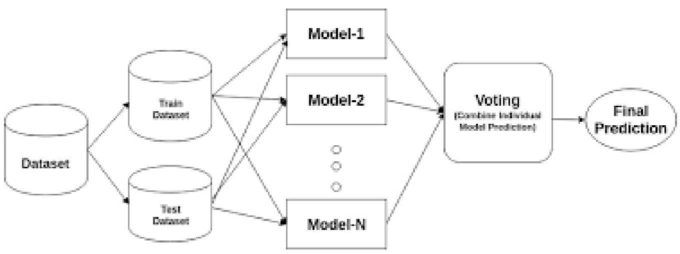


7. Neural Nets: It is a technique or model whose working is inspired by the human brain. It is a highly interconnected network of nodes in a structure resembling layers.[6][10]

The major use of neural nets model is in medical diagnosis, targeted marketing by social networking and financial predictions by processing historical data.[5]



8. Ada-boost model: It is a boosting technique used as an ensemble method. Works on a principle of learners growing sequentially. Each preceding learner grows from the previous learner, in other words we can say that weak grows into strong ones. In the working suppose n decision trees are in the model, then first one is given priority and sent as input in other models and this is how the weak turns to strong.



6. Results and Observations

After calculating the accuracy of all the above-mentioned models, it was observed that Random Forest Classifier had the best accuracy of 81.41 % for the given dataset, followed by

- K-Nearest Neighbor classifier - Accuracy 80.58%
- Support Vector Machine algorithm - Accuracy 75.41%
- Logistic Regression - Accuracy 69.77%
- Neural Networks - Accuracy 67.84%
- Stochastic Gradient Descent - Accuracy 65.53%
- Decision Tree Classifier - Accuracy 65.10%
- Naive Bayes Classifier - Accuracy 51.95%
- Adaboost algorithm - Accuracy 46.28%

100	6	61	12	1	19	85	0	12	23
0	275	1	0	0	21	0	0	1	10
20	11	166	25	4	8	22	11	10	9
13	4	30	102	13	2	57	46	17	17
8	1	15	36	109	0	44	60	36	2
10	59	34	6	0	117	8	13	5	34
5	0	3	11	3	0	269	0	3	9
0	0	17	18	2	1	0	208	13	8
19	1	61	10	23	3	6	29	149	15
8	6	49	40	2	7	96	13	17	62

Figure: Naive Bayes Classifier Confusion Matrix

257	9	9	3	8	15	12	0	6	0
1	302	3	0	0	2	0	0	0	0
51	7	150	12	7	29	0	13	13	4
15	10	19	145	62	7	2	25	14	2
15	5	8	2	237	0	6	15	21	2
17	48	6	1	2	203	0	2	7	0
32	1	5	13	12	0	231	0	5	4
1	2	7	4	19	4	0	221	7	2
21	2	15	8	61	7	3	4	192	3
76	13	30	46	25	29	11	14	30	26

Figure: Stochastic Gradient Descent Confusion Matrix

256	5	22	4	2	8	6	0	10	6
0	297	1	0	0	10	0	0	0	0
9	2	230	11	1	13	0	1	12	7
7	6	8	246	8	0	2	4	6	14
7	1	15	24	217	0	9	21	14	3
2	30	16	2	0	233	0	2	1	0
6	0	2	15	5	0	252	0	4	19
0	0	6	17	2	2	0	229	10	1
3	1	16	20	13	0	2	11	248	2
11	1	20	35	0	7	7	2	10	207

Figure: K-Nearest Neighbor Confusion Matrix

169	4	34	14	11	15	21	5	18	28
3	258	15	0	0	23	1	3	1	4
31	6	147	13	11	17	13	13	10	25
10	2	16	165	25	8	8	20	16	31
5	0	13	25	211	5	14	15	20	3
14	16	13	7	1	207	1	13	2	12
6	0	8	15	9	2	240	2	5	16
2	1	10	14	11	6	1	207	9	6
13	3	18	20	26	3	6	13	195	19
19	7	24	22	15	17	24	5	15	152

Figure: Decision Tree Classifier Confusion Matrix

246	1	31	8	4	8	15	0	6	0
0	294	1	0	0	12	0	0	0	1
13	1	212	12	0	23	3	3	9	10
1	4	11	235	14	1	5	8	10	12
4	0	7	8	250	2	13	15	10	2
7	21	5	1	0	251	0	0	1	0
1	0	1	2	3	1	281	0	6	8
0	0	12	4	2	1	0	238	5	5
5	2	9	11	7	4	4	17	254	3
7	2	29	34	1	12	19	4	13	179

Figure: Random Forest Classifier Confusion Matrix

241	5	10	9	2	12	24	0	7	9
0	294	2	0	0	10	0	0	0	2
19	0	192	18	0	23	1	5	5	23
2	6	7	199	18	7	10	12	14	26
7	2	7	18	220	0	8	16	28	5
12	25	9	2	0	234	0	1	2	1
10	0	0	7	4	0	270	0	2	10
0	1	10	8	4	1	0	233	4	6
13	2	16	18	20	3	5	10	220	9
19	2	24	45	7	11	16	9	10	157

Figure: Support Vector Machine Confusion Matrix

212	5	12	7	4	18	40	0	11	10
2	287	5	0	0	10	0	0	0	4
23	1	166	18	2	21	1	19	9	26
3	5	15	182	23	3	16	21	15	18
8	2	8	17	201	0	15	24	31	5
11	28	14	5	1	214	0	2	7	4
10	0	0	7	4	0	266	0	2	13
0	0	14	8	11	0	0	226	3	5
14	1	19	12	29	5	8	10	210	8
35	3	25	32	5	14	22	18	19	127

Figure: Logistic Regression Confusion Matrix

194	2	52	3	4	16	22	0	13	13
3	284	3	1	0	16	0	0	1	0
24	0	151	14	1	33	0	6	17	40
5	4	11	161	29	4	9	21	20	37
7	0	5	22	202	4	12	17	37	5
12	25	20	0	0	212	1	2	2	12
17	0	0	9	5	0	261	0	0	11
0	0	2	8	5	0	0	236	13	3
5	1	25	16	30	1	1	11	223	3
25	2	51	46	2	18	20	12	15	109

Figure: Neural Network Confusion Matrix

110	2	23	17	7	10	122	0	19	9
3	283	2	1	1	12	0	0	1	5
49	8	79	10	37	10	18	35	20	20
11	7	6	77	23	0	58	86	19	14
8	3	7	37	73	1	93	65	20	4
29	60	16	14	11	92	11	27	13	13
4	0	2	5	3	0	283	0	0	6
0	1	6	5	6	0	0	242	6	1
25	5	25	30	40	0	22	60	103	6
31	4	23	42	10	6	72	55	12	45

Figure: Adaboost Algorithm Confusion Matrix

Once we have determined that Random Forest Classifier is the machine learning algorithm most suitable for our chosen GTZAN dataset, we plotted the confusion matrix for the same, taking into account all the genres that were taken into consideration in our GTZAN dataset.

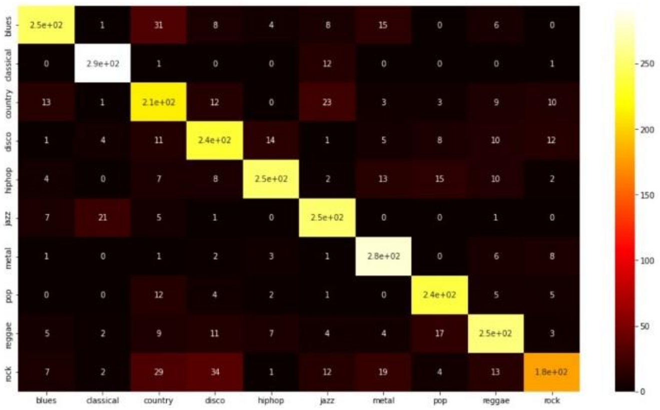


Figure: Random Forest Classifier Confusion Matrix

7. Conclusion

To determine which machine learning algorithm is most suitable for the purpose of this research i.e., to determine the genre of a music/audio file, we calculated the accuracies of 9 different machine learning algorithms.

Our research indicates that Random Forest Classifier (Accuracy - 81.41%) caters to our needs the most followed by K-Nearest Neighbor Classifier (Accuracy - 80.58%) and hence, these two algorithms are most suitable for Genre classification on the GTZAN dataset.

8. References

1. R.Thiruvengatanadhan, 'Music Genre Classification using SVM', International Research Journal of Engineering and

Technology (IRJET).
2. Ardiansyah*, Boy Yuliadi**, Riad Sahara***, “Music Genre Classification using Naïve Bayes Algorithm”, International Journal of Computer Trends and Technology (IJCTT) – Volume 62 Number 1 – August 2018.
3. Nilesh M. Patil1, Dr. Milind U. Nemade, “Music Genre Classification Using MFCC, K-NN and SVM Classifier”, International Journal of Computer Engineering In Research Trends.
4. Ahmet Elbir1, Hilmi Bilal Çam2 , Mehmet Emre İyican2 , Berkay Öztürk2 , Nizamettin Aydın1 , “Music Genre Classification and Recommendation by Using Machine Learning Techniques”,
5. Saad ALBAWI, Tareq Abed MOHAMMED, “Understanding of a Convolutional Neural Network”, 2017 International Conference on Engineering and Technology (ICET).
6. N. Pelchat and C. M. Gelowitz, "Neural Network Music Genre Classification," in Canadian Journal of Electrical and Computer Engineering, vol. 43, no. 3, pp. 170-173, Summer 2020, doi: 10.1109/CJECE.2020.2970144.
7. Music Genre Classification using Machine Learning Algorithms: A comparison Snigdha Chillara1, Kavitha A S2, Shwetha A Neginhal3, Shreya Haldia4, Vidyullatha K S5 International Research Journal of Engineering and Technology (IRJET)
8. Y. Huang and L. Li, "Naive Bayes classification algorithm based on small sample set," 2011 IEEE International Conference on Cloud Computing and Intelligence Systems, 2011, pp. 34-39, doi: 10.1109/CCIS.2011.6045027.
9. M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines," in IEEE Intelligent Systems and their Applications, vol. 13, no. 4, pp. 18-28, July-Aug. 1998, doi: 10.1109/5254.708428.
10. N. Jmour, S. Zayen and A. Abdelkrim, "Convolutional neural networks for image classification," 2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET), 2018, pp. 397-402, doi: 10.1109/ASET.2018.8379889.
11. Yin, Qiwei & Zhang, Ruixun & Shao, XiuLi. (2019). CNN and RNN mixed models for image classification. MATEC Web of Conferences. 277. 02001. 10.1051/matecconf/201927702001.
12. [13] Guo, Gongde & Wang, Hui & Bell, David & Bi, Yaxin. (2004). KNN Model-Based Approach in Classification.
13. [14] M. A. Hossan, S. Memon and M. A. Gregory, "A novel approach for MFCC feature extraction," 2010 4th International Conference on Signal Processing and Communication Systems, 2010, pp. 1-5, doi: 10.1109/ICSPCS.2010.5709752.
14. [15] FEATURE EXTRACTION USING MFCC Shikha Gupta1 , Jafreezal Jaafar2 , Wan Fatimah wan Ahmad3 and Arpit Bansal4 Signal & Image Processing : An International Journal (SIPIJ) Vol.4, No.4, August 2013