International Journal of Database Theory and Application
Vol.9, No.5 (2016), pp.129-140
http://dx.doi.org/10.14257/ijdta.2016.9.5.13

Automatic Tagging of Songs Using Machine Learning

Anurag Das[#1], Rajat Bhai[#2], Shaiwal Sachdev[#3], Tanushree Anand[#4] and Utkarsh Kumar[#5]

[#]*Undergraduate students, IIIT Allahabad, India*
[1]*IIT2013198@iiita.ac.in*, [2]*IIT2013030@iiita.ac.in*, [3]*IIT2013196@iiita.ac.in*, [4]*IIT2013192@iiita.ac.in*, [5]*IRM2013002@iiita.ac.in*

Abstract

In this research work automatic tagging of songs using machine learning has been performed so that searching could be made effective while selecting songs. The goal of this research paper is to propose a system that will automatically recognize the genre of the tracks and tag them respectively by using different parameters obtained by acoustic analysis. The research work utilizes different combinations of algorithms and music parameters to accurately classify tracks into genres.

Keywords*: Automatic Tagging, Genre Classification, Machine Learning*

1. Introduction

People have different taste in music and different people like to listen different genres (so called tags like rock, metal, folk, jazz etc) of music. So, there is a need that the songs must be classified in accordance with the tags that they represent which would be beneficial for both listener and music companies (for organizing music in a better way). Thus, songs must have searchable tags so that it could be easily recognizable. These types of tags are called metadata.

Tags can be assigned to tracks in three general ways:

- Group of experts manually tagging the tracks. This is very time consuming but accurate.
- Users doing this manually will make the result inconsistent as different people have different perception and feeling about same genre.
- Automatic Tagging by using the data obtained from acoustic analysis

The goal of this research paper is to propose a system that will automatically recognize the genre of the tracks and tagging them respectively by using the different parameters obtained by acoustic analysis. The research work utilizes different combinations of algorithms and music parameters to accurately classify tracks into genres.

2. Related Work

Many studies have been made in the past related to automatic tagging and authors have used different combinations of algorithms and acoustic parameters to achieve results with different accuracies. Both low level features and high level symbolic features can be used to accomplish genre classification [1]. Low level features describes the characteristic of audio signal and establish how a human ears perceives the music while the high level features estimates the musical elements such as pitch and tempo. Low level features generally include Mel Frequency Cepstral Coefficients (MFCC). Each MFCC is a set of coefficients describing a segment of audio sample typically less than one second. MFCC is therefore used in many tagging application and also speech recognition systems. Multiple MFCCs are also used to represent a whole track. High level features generally include pitch and loudness [2-3].

ISSN: 2005-4270 IJDTA

Salamon *et al*. published a research work in 2012, where a comparison between high-level and low-level features and a combination of those were evaluated. The pitch, the vibrato and their duration were used as high-level features, and the MFCC was used as a low-level feature. The algorithms used were, Support Vector Machine (SVM), Random Forest (RF), K-Nearest Neighbours (KNN) and Bayesian Network (BN) [4].

Liang *et al*., in his research work used different combinations of features available in MSD, including timbre (which is unique for MSD), tempo, loudness and a bag- of-word feature (derived from the lyrics) [5]. The accuracy seemed to be also dependent on the size of dataset used. The accuracy obtained by those who used the full dataset was more than those who used the subset. In summary, all of the above mentioned studies used quite similar approaches, *i.e*. all used the MFCC feature, except for Liang *et al*. which used the corresponding timbre feature from MSD, to base the classification upon.

All studies used the supervised machine learning for classification into different tracks. Most of the studies used MFCC as the low level feature and additional high level features as pitch, loudness, key, *etc.* were used to improve the result. There are several research work available in which music files are used to classify the emotions on the basis of various low level and high level features [6-7].

3. Proposed Methodology

Figure 1 represents the steps of proposed methodology.

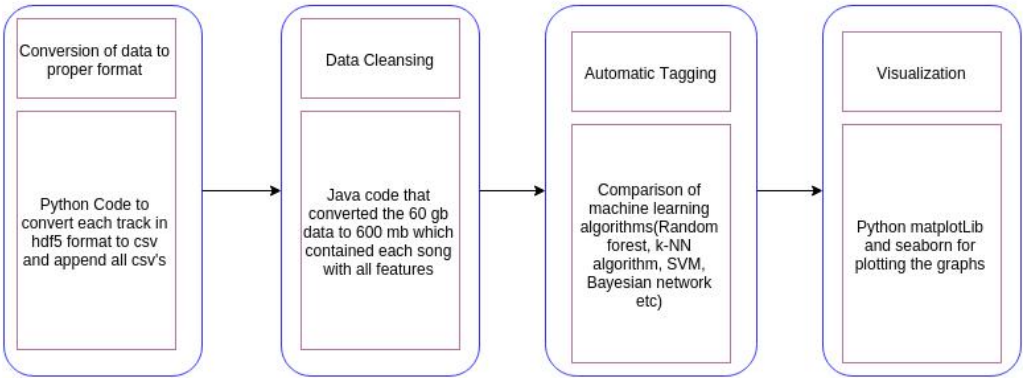


Figure 1. Proposed Methodology

3.1 Bringing Data to Proper Format

The data which was in hdf5 format is converted to csv format which accounts to approximate 2 GB which is then converted to approx 60 GB csv data with the help of python code (open source code given by Infobright Inc. [8]). The python code converts each single file/track to csv and appends all the files in same csv. The final obtained csv file has about 427 fields.

3.2 Data Cleansing

Preprocessing is an important step while dealing with large set of data having multiple features [9]. The csv file obtained after conversion is sent to the java codes for further processing. The current file has 427 fields and is converted to a csv file with 18 fields. A lot of challenges were faced during the conversion. For example, the address field had comma itself in the csv file comma separated file so processing and finding this error was a big menace. Firstly we removed unnecessary fields such as section start, similar artist, song id, release7digitalid *etc.* from this dataset. Then we had Boolean fields for genre's so

we had to convert those to a single field. After all of this conversion, we got a single csv file with 18 fields. The snapshots of the work are given in subsequent sections.

3.3 Automatic Tagging Using Machine Learning Algorithms

For each track there is a set of metadata such as the name of artist, title of the track, recording year and the acoustic features (pitch, timber, loudness). The metadata of each track contains the possible list of genres that track can relate as an estimation of genres connected to each track and here highest frequency range can be used for the same. So we have 9 major genre classifications as indicated below:

i. Rock
ii. Pop
iii. Electronics Dance Music
iv. Jazz
v. Vocals
vi. Hip-hop
vii. Folk
viii. Instrumental
ix. Soul

We will classify each track into these 9 classes and hence generate test data. There are different parameters which differ from the different genres of the songs such as tempo, timber, pitch, beat so we can use the different combination of these to teach our learning machine to differentiate between genres. Here are some chosen combinations of the data.

- Mean of Timbre and Standard Deviation of Timbre.
- Mean of Pitch and Standard Deviation of Pitch.
- Mean of Timbre, Standard Deviation of Timbre, Tempo, Key, Loudness
- Mean of Pitch and Standard Deviation of Pitch, Mean of Timbre and Standard Deviation of Timbre, Tempo, Key, Loudness
- Mean and standard deviation of Timbre, Pitch and Tempo, Key, Key confidence, Loudness, Mode
- Tempo, Key, Loudness

The prediction of which algorithm to use in accurate prediction of the result is rather difficult. Hence a set of classification algorithms are chosen. We apply these algorithms on our dataset and the most accurate one will be used to predict the final result. The selected algorithms are as follows:

3.3.1 Bayesian Networks

Bayesian networks are graphical representation of probabilistic relationship between random events. Each node in the graphical model is a random variable and all nodes are conditionally independent except those which share an edge between them, *i.e*. each node is conditionally dependent only on its parent nodes. The Bayesian networks are directed graphs. Learning through Bayesian network is based on some prior knowledge of the events. They are generally used for probabilistic inference of events. More generally bayes rule is applied to find the probability at each node keeping the independence of random variables in mind. They can also be used for classification. The probability obtained at the output states with respect to certain threshold value determines the class of input [10-11].

3.3.2 Random Forests

Random Forest is based on ensemble learning. It uses random decision forests for classification. There is a set of features whose subset is randomly chosen again and again for generating different decision trees. This brings randomness in the decision making

process. Also while selecting each node we introduce randomness in selecting the nodes. It is also effective in removing noisy input as it uses a subset of features and thus it does contains some decision trees which do not contain the noisy elements. The results from all the decision trees decide the final classification [12].

3.3.3 K-Nearest Neighbors

It is the simplest classification algorithm which stores all the available cases and predicts the class for a new item based on the similarity measure. One way to find the similarities of a new item by taking the votes of the nearest neighbors. There are many nearest neighbors possible but we can fix their number up to k *i.e*. we will consider the neighbors with the first k distances only. For the calculation of the nearest neighbors various distance measures can be taken into account for example Euclidean distance, Manhattan distance or Minkowski distance. The new item is said to belong to the class which resembles most of the neighbors. In case of a tie any class can be chosen [13].

3.3.4 Support Vector Machine

Support Vector Machine is the supervised learning algorithm that takes labeled training data as input and produces an optimal hyper plane which can categories the new item. According to the SVM theory it tries to find out the largest minimum distance among the training examples. The linear and non-linear classifiers are used for the classification of the data [14-15].

3.4 Experimental Analysis

3.4.1 Dataset Description

The dataset used in this project is The Million Song Dataset (MSD) [16]. This set contains a large amount of pre analyzed tracks. MSD contains 1,000,000 tracks by 44,745 unique artists and the subset contains 10,000 tracks by 3,888 unique artists. For current analysis we are using the subset which contains 10,000 tracks of Size of 2 GB in hdf5 format. It has been be provided by The Echo Nest [17] and LabROSA [18]. Each song represents an hdf5 file with approximately 50 fields. This shows the view of single track in hdf5 format.

The fields are divided in following three parts

- Analysis: It contains data about the individual track.
- Metadata: It contains metadata about the track.
- Musicbrainz: Additional information provided by Musicbrainz Company.

end of fade in	float	seconds at the beginning of the song
energy	float	energy from listener point of view
key	int	key the song is in
key confidence	float	confidence measure
loudness	float	overall loudness in dB
mode	int	major or minor
mode confidence	float	confidence measure
release	string	album name
release 7digitalid	int	ID from 7digital.com or -1
sections confidence	array float	confidence measure
sections start	array float	largest grouping in a song, e.g. verse
segments confidence	array float	confidence measure
segments loudness max	array float	max dB value
segments loudness max time	array float	time of max dB value, i.e. end of attack
segments loudness max start	array float	dB value at onset
segments pitches	2D array float	chroma feature, one value per note
segments start	array float	musical events, ~ note onsets
segments timbre	2D array float	texture features (MFCC+PCA-like)
similar artists	array string	Echo Nest artist IDs (sim. algo. unpublished)
song hotttnesss	float	algorithmic estimation
song id	string	Echo Nest song ID
start of fade out	float	time in sec
tatums confidence	array float	confidence measure
tatums start	array float	smallest rythmic element
tempo	float	estimated tempo in BPM
time signature	int	estimate of number of beats per bar, e.g. 4
time signature confidence	float	confidence measure
title	string	song title
track id	string	Echo Nest track ID
track 7digitalid	int	ID from 7digital.com or -1
year	int	song release year from MusicBrainz or 0

Figure 2. Track Information in Hdfview and Related Fields

3.4.2 Tools and Languages Used

Following tools have been used in this research work:

Scikit Learn

It is an open source machine learning, data mining and data analysis library for python. It is build over NumPy, SciPy and matplotlib library of python. It has algorithms from all domains of machine learning like regression, classification, clustering, dimensionality reduction *etc*. The algorithms are general and easy to use.

HdfViewer

It is a great tool for visualizing and editing HDF5 and HDF4 format files. The viewer helps in viewing the file hierarchy in tree structure, create new files, add or delete groups and datasets, add and delete and modify the attributes and to view and modify the content of dataset. This HDF (Hierarchical Data Format) format results in high compression of data [19].

Matplotlib

It is a library in python used for plotting and producing quality figures. The plot generated can be saved in a variety of formats. Pyplot provides an interface to the matplotlib library and has functions which have close resemblances to that of MATLAB.

Anaconda

Anaconda is completely free python distribution. It includes various popular python packages for science, math, engineering and data analysis. It contains more than 400 packages. The name of some important packages are NumPy, Pandas, SciPy, Matplotlib and many more. Anaconda is available for Linux, Windows and OS X and is free and open source.

LaTeX

LaTeX is superb tool for creating scientific and technical documents. It has several features which include defining project, including chapters, add new section, typesetting of complex mathematical functions. Also the bibliography, references and index are automatically generated. Following languages has been used in this research work.

Python

Python is a high level, general purpose and dynamic programming language. It supports multiple programming paradigms including object-oriented, imperative, functional and procedural. The language is available for the various operating systems. It is free and open source and has a community based development model. Using third party tools such as Py2exe or Pyinstaller python code can be packaged into standalone executable programs.

Java

It is high level programming language which is concurrent, class based and object oriented. It promotes the idea of write once, run anywhere (WORA) methodology. Java applications are compiled to bytecode that can run on any JVM (Java Virtual Machine) regardless of the architecture of the computer. The language derives much of its syntax from C and C++.

3.4.3 Snapshots of the Experimental Analysis

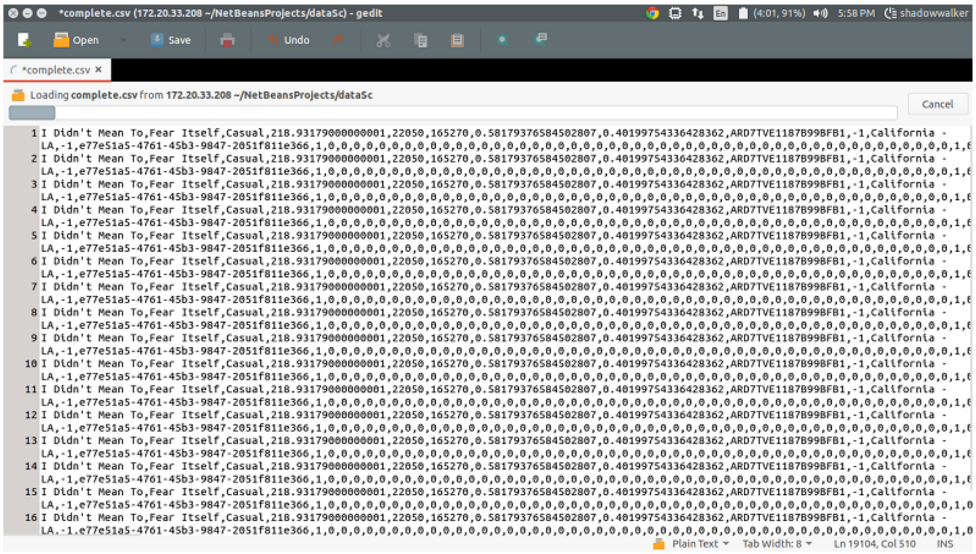


Figure 3. Complete Dataset in Gedit

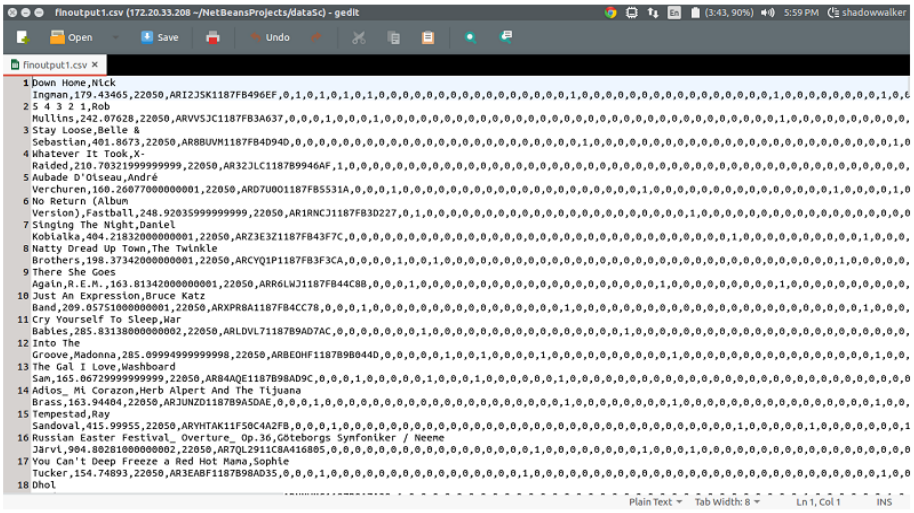


Figure 4. Cleaned Data

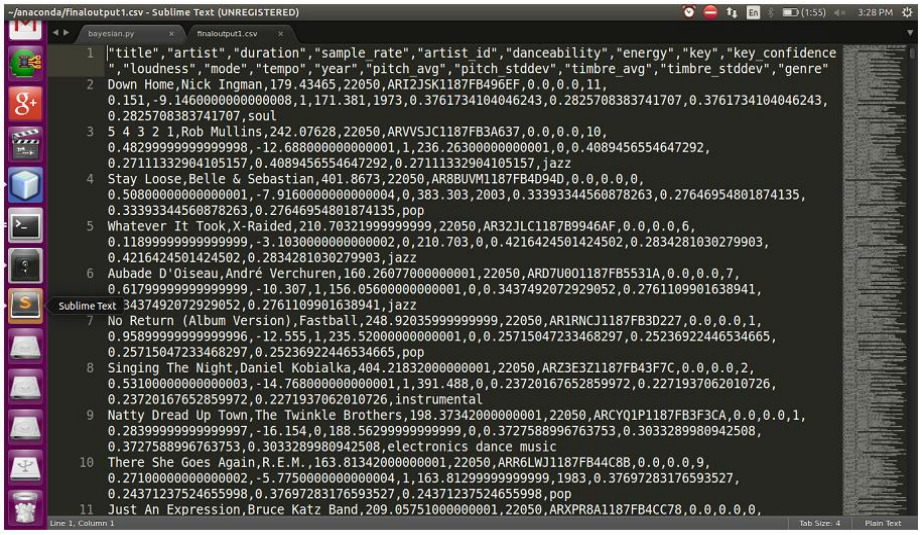


Figure 5. Reduced Data with Required Features

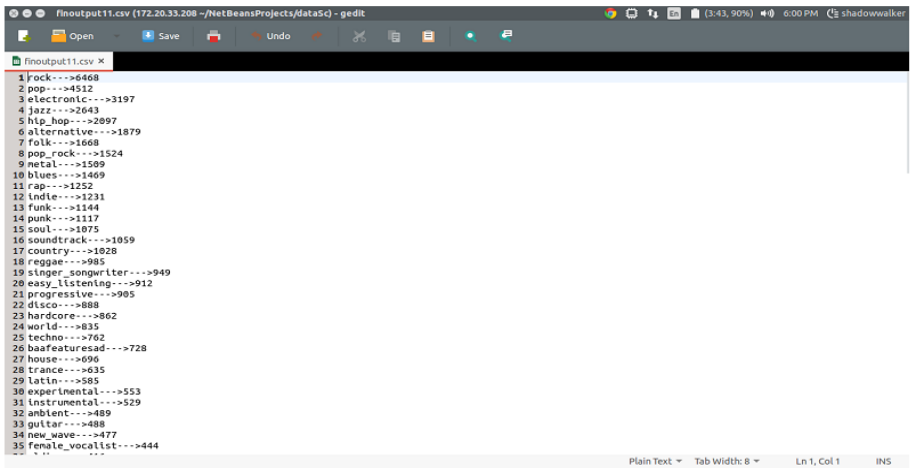


Figure 6. Frequency of Songs Obtained with Java Code

3.4.4 Assigning Single Tag to Multi Tags

We had 133 tags available from the dataset given by Million Song Dataset which was converted to 9 later as tags were overlapping.

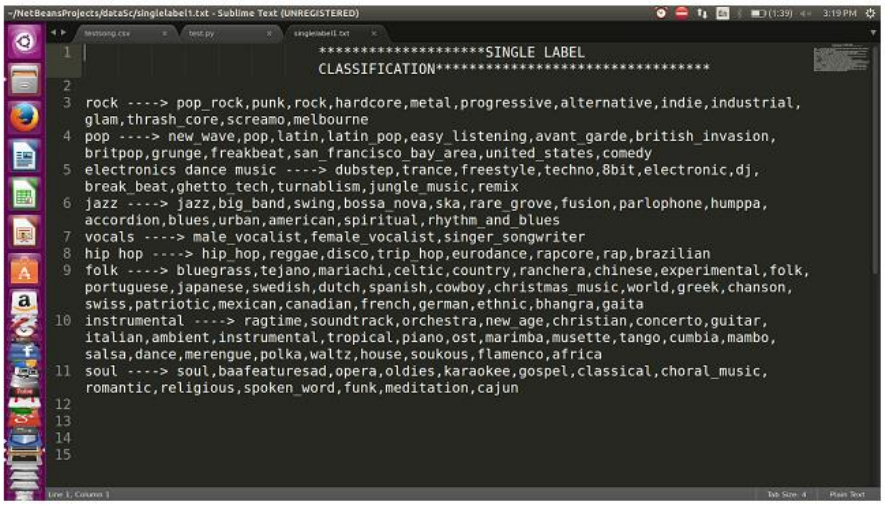


Figure 7. Single Label Classification

3.4.5 Automatic Tagging Using Classification Algorithm

We fed external song to the Echonest API which returned the desired features for the machine learning algorithms to run on. The results obtained are given below:-

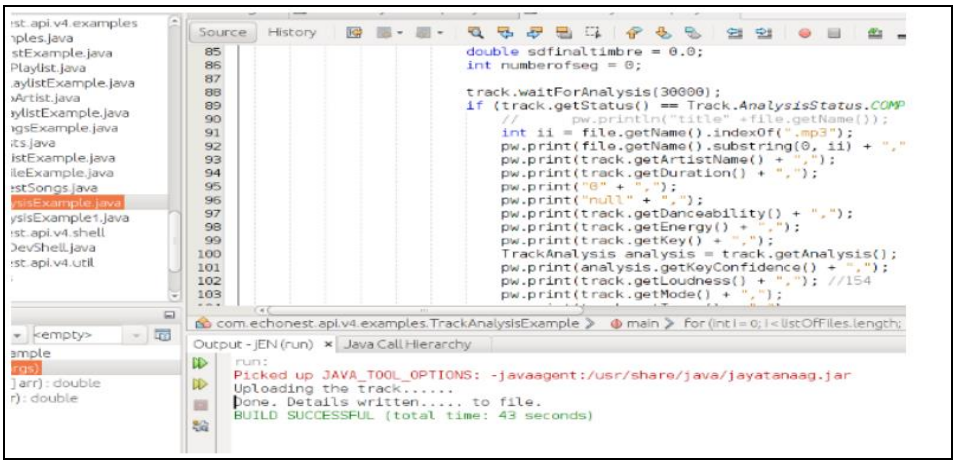


Figure 8. Running on External Mp3 Song

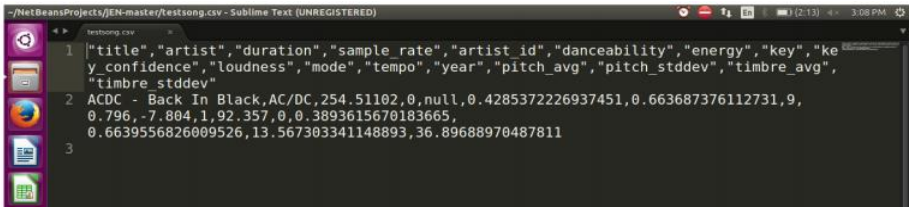


Figure 9. Output Obtained from the API

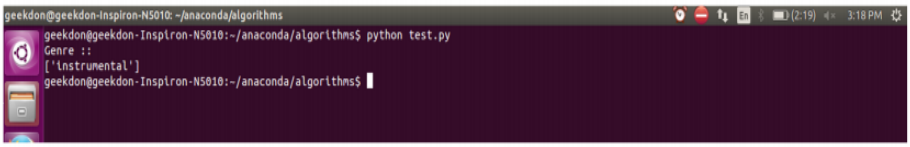


Figure 10. Tag Obtained for the Given Song

3.4.5 Results of Classification Algorithms Used for Automatic Tagging

Algorithms	Combination of features	Accuracy
KNN	Mean of Timbre and Standard Deviation of Timbre	33%
	Mean of Pitch and Standard Deviation of Pitch	28%
	Mean of timbre, Standard Deviation of Timbre, tempo, key, loudness	36%
	Mean of Pitch and Standard Deviation of Pitch, Mean of Timbre and Standard Deviation of Timbre, tempo, key, loudness	35%
	Mean and standard deviation of Timbre, Pitch and tempo, key confidence, loudness, mode	38%
	Tempo, key, loudness	28.58%
Naïve Bayes	Mean of Timbre and Standard Deviation of Timbre	40%
	Mean of Pitch and Standard Deviation of Pitch	42%
	Mean of timbre, Standard Deviation of Timbre, tempo, key, loudness	44%
	Mean of Pitch and Standard Deviation of Pitch, Mean of Timbre and Standard Deviation of Timbre, tempo, key, loudness	42%
	Mean and standard deviation of Timbre, Pitch and tempo, key, key confidence, loudness, mode	47%
	Tempo, key, loudness	37%
Algorithms	**Combination of features**	**Accuracy**
Random Forest	Mean of Timbre and Standard Deviation of Timbre	42%
	Mean of Pitch and Standard Deviation of Pitch	40%
	Mean of timbre, Standard Deviation of Timbre, tempo, key, loudness	43%
	Mean of Pitch and Standard Deviation of Pitch, Mean of Timbre and Standard Deviation of Timbre, tempo, key, loudness	44%
	Mean and standard deviation of Timbre, Pitch and tempo, key, key confidence, loudness, mode	46%
	Tempo, key, loudness	39%
Support Vector Machine	Mean of Timbre and Standard Deviation of Timbre	44%
	Mean of Pitch and Standard Deviation of Pitch	41%
	Mean of timbre, Standard Deviation of Timbre, tempo, key, loudness	45%
	Mean of Pitch and Standard Deviation of Pitch, Mean of Timbre and Standard Deviation of Timbre, tempo, key, loudness	46%
	Mean and standard deviation of Timbre, Pitch and tempo, key, key confidence, loudness, mode	48%
	Tempo, key, loudness	35%

Table 1. Comparison of Machine Learning Algorithms Used

No.	k-NN	Naive Baiyes	Random Forest	SVM
1.	33.0	40.0	42.0	44.0
2.	28.0	42.0	40.0	41.0
3.	36.0	44.0	43.0	45.0
4.	35.0	42.0	44.0	46.0
5.	38.0	47.0	46.0	48.0
6.	28.58	37.0	39.0	35.0

3.4.6 Data Visualization

The entire aim of the project is to enable automatic genre tagging of music from varied sources. Interactive Visualization of the analyzed data is extremely important to foster the Dialogue between the data analysts and decision makers at all levels (users, music industry *etc.*).

i. It enables them to grasp analytical results presented visually and relevance among the millions of variables, communicated concepts and hypotheses to others, and even predict the future.

ii. It help you understand trends, patterns, and to make correlations.

For Data Visualization, we have used the matplotLib package and seaborn library of python.

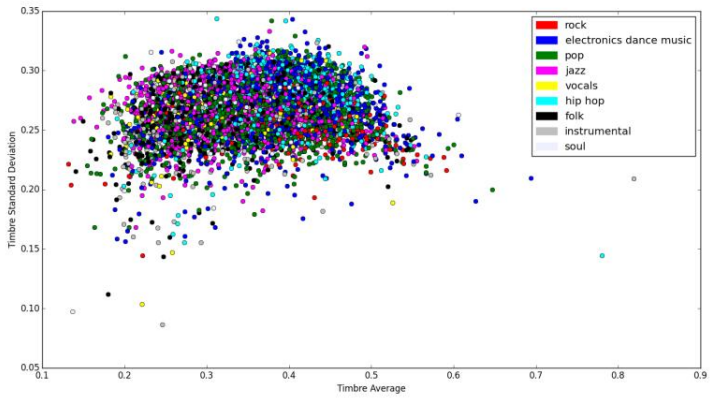


Figure 12. Song Distribution Based on Timbre Average and Standard Deviation

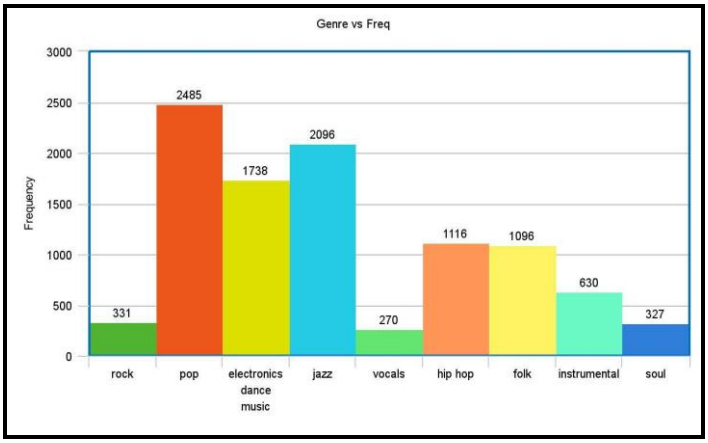


Figure 13: Genre vs. Frequency of Tags

4. Conclusion

In present research work, we were able to predict and assign the tags to the songs with maximum accuracy of 48 percent using Support Vector Machines with feature combination as mean and standard deviation of Timbre, Pitch and tempo, key, confidence, loudness and mode. There are various research works available in which the same million song dataset has been classified and we found the maximum accuracy achieved for classification was 51 percent. In present research work, although the accuracy obtained is less but with more number of class labels as tags. Previous research was based on 6 class labels as tags but we successfully extended the class labels up to 9 genres for better classification of songs with slightly less efficiency. We were not able to compromise on the number of tags to generate irrelevant result and to get higher accuracy.

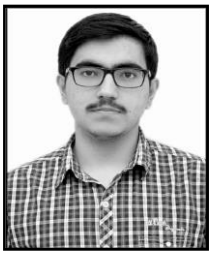
References

[1]. C. C. M. Yeh, J. C. Wang, Y. H. Yang and H. M. Wang, "Improving music auto-tagging by intra-song instance bagging", Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, Florence, doi: 10.1109/ICASSP.2014.6853977, **(2014)**, pp. 2139-2143.
[2]. Y. H. Yang, "Towards real-time music auto-tagging using sparse features", Multimedia and Expo (ICME), 2013 IEEE International Conference on, San Jose, CA, **(2013)**, pp. 1-6.
[3]. K. Ellis, E. Coviello, A. B. Chan and G. Lanckriet, "A Bag of Systems Representation for Music Auto-Tagging", in IEEE Transactions on Audio, Speech, and Language Processing, vol. 21, no. 12, pp. 2554-2569.
[4]. S. Justin, R. Bruno and G. Emilia, "Musical Genre Classification Using Melody Features Extracted From Polyphonic Music Signals", In Proceeding IEEE Int. Conf. on Acoustic., Speech and Signal Process. (ICASSP), **(2012)**.
[5]. L. Dawen, G. Haijie and O. C. Brendan, "Music Genre Classification with the Million Song Dataset", Pittsburgh, Carnegie Mellon University, **(2011)**.
[6]. D. Tomar and S. Agarwal, "Multi-label Classifier for Emotion Recognition from Music", Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics, Springer, India, **(2016)**, pp. 111-123.
[7]. D. Tomar, D. Ojha and S. Agarwal, "An emotion detection system based on multi least squares twin support vector machine", Adv. in Artif. Intell. 2014, Article 8, **(2015)**.
[8]. "Infobright Developed by: Infobright Intern Team", Author: Infobright Intern Team Version 0.1, **(2011)**.
[9]. D. Tomar and S. Agarwal, "A survey on pre-processing and post-processing techniques in data mining", International Journal of Database Theory &; Application, vol. 7, no. 4, **(2014)**.
[10]. D. Heckerman, "A Tutorial on Learning with Bayesian Networks", Learning in Graphical Models, MIT Press, Cambridge, MA, **(1999)**.
[11]. Z. Barutcuoglu, R. E. Schapire and O. G. Troyanskaya, "Hierarchical Multi-label Prediction of Gene Function", Bioinformatics, January, **(2006)**.
[12]. R. S. Mark, "Machine Learning Benchmarks and Random Forest Regression", Division of Biostatistics, University of California, **(2003)**.
[13]. L. Tao, M. Ogihara and Q. Li, "A Comparative Study on Content-Based Music Genre Classification", Special Interest Group on Information Retrieval, http://users.cis.fiu.edu/~taoli/pub/sigir03-p282-li.pdf, **(2003)**, pp. 282.
[14]. C. Cortes and V. Vapnik, "Support-vector networks", Machine Learning, **(1995)**.
[15]. A. S. Divya and G. N. Pandey, "SVM based context awareness using body area sensor network for pervasive healthcare monitoring", In: Proceedings of the first international conference on intelligent interactive technologies and multimedia. ACM, **(2010)**, pp. 271–8.
[16]. "Infobright Developed by: Infobright Intern Team", Author: Infobright Intern Team http://github.com/infobright/MillionSong, **(2011)**.
[17]. The Echo Nest http://the.echonest.com (Viewed: 2015-09-01)
[18]. Dataset downloaded from website http://labrosa.ee.columbia.edu/millionsong/.
[19]. S. Agarwal and B. R. Prasad, "Comparative Study of Big Data Computing and Storage Tools: A Review", International Journal of Database Theory (IJDTA), vol. 9, no. 1, **(2016)**, pp. 45-66.

Authors

Anurag Das, He is an undergraduate student at Indian Institute of Information Technology, Allahabad. His research interests are in Machine Learning, Big Data Analysis and Computer vision.

Rajat Bhai, He is currently an undergraduate student at Indian Institute of Information Technology, Allahabad. Currently he is working on project "multivariate cryptography", which is concerned with the designing of a cryptosystem that can handle the attacks made by quantum computers.



Shaiwal Sachdev, He is an undergraduate student of Information Technology of Indian Institute of Information Technology (IIIT), Allahabad, India. His primary research interests are Machine Learning, Data Mining and Neural Networks.

Tanushree Anand, She is pursuing Bachelor of Technology Degree in Information Technology (IT) at Indian Institute of Information Technology, Allahabad. Her primary research interests are Big Data Analysis, Data Mining, Digital Image Processing and Large Volume Visualization.

Utkarsh Kumar, He is an undergraduate student at Indian Institute of Information Technology, Allahabad. His research interests are in Machine Learning, Big Data Analysis and Computer vision.