MUSIC ANALYSIS USING DATA MINING TECHHNIQUES

School of Information Technology and Engineering

A PROJECT REPORT

for

DATA MINING TECHNIQUES (ITE2006)

In

B.Tech (Information Technology)

By

Koonamneni Janavi (20BIT0248)

Makam Ranga Rakshith (20BIT0360)

Ambarapu Rahithya Teja (20BIT0296)

Fall Semester 2022 - 23

Under the Guidance of

Prof. B. VALARMATHI

Associate Professor, SITE

DECLARATION BY THE CANDIDATE

We here by declare that the project report entitled **"MUSIC ANALYSIS USING DATA MINING TECHHNIQUES"** submitted by us to Vellore Institute of Technology University, Vellore in partial fulfillment of the requirement for the award of the course **Data Mining Techniques (ITE2006)** is a record of bonafide project work carried out by us under the guidance of **Prof. B.Valarmathi.** We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other course.

Place : Vellore

Signature

Date : 18-10-2022.

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology & Engineering [SITE]

<u>CERTIFICATE</u>

This is to certify that the project report entitled **"MUSIC ANALYSIS USING DATA MINING TECHHNIQUES"** submitted by

Koonamneni Janavi (20BIT0248)

Makam Ranga Rakshith (20BIT0360)

Ambarapu Rahithya Teja (20BIT0296)

to Vellore Institute of Technology University, Vellore in partial fulfillment of the requirement for the award of the course **Data Mining Techniques (ITE2006)** is a record of bonafide work carried out by them under my guidance.

Prof. B.Valarmathi

GUIDE

Associate Professor (Senior), SITE

ABSTRACT

Today, several music platforms provide thousands of new songs; it becomes ambitious to find highly appealing or like-able songs within such loads of data. In our project we have considered various attributes like acousticness, danceability, energy, instrumentalness, key, loudness, mode, speechness, valence and artist to find whether the audience will like the song or not. “1” is used to denote that the audience will like the song and “0” is used to denote that they won’t like that song. We have built a classifier that could predict whether the audience will like the song. Whole data is divided into two parts. 80% of the data is used as training data and the rest 20% as test data.

Keywords –

Data visualization, Visual analytics, Music, Data analysis, Text mining.

I.Introduction:

We have used three algorithms to build a classifier that predicts whether the audience will like the song or not. We have used algorithms like Decision Tree Algorithm, Random Forest and Multiple Models to predict whether a song will be liked by the user or not. We calculated the accuracy and error percentage of the different algorithms by comparing the calculated output with the value in the database and got the best results for accurate model. We compared the mentioned algorithms based on their results for a database which comprised of 42000 songs from Spotify with 16 different attributes and an output as 1 if the audience might like that particular song and 0 if the audience would not like that particular song.

II.Background:

Music is the one of the most trending in present days. All of us love to listen music but, there are different perceptions or expectations or opinion on the particular music whether a user like’s it or not. In this project we are going to develop a model which helps us to predict whether a music will liked or not liked by people. For, these we are going to choose dew algorithms to check which gives the more accuracy on prediction.

III.Objectives:

The goal of this project is to predict the music by analysing the data set using machine learning classifiers like decision tree classifier, Random forest classifier, Boosting classifiers, Ensemble classifier.

IV.Literature Survey:

S.No.	Title of the paper and year	Algorithms used	Data set being used	Performance measures	Scope for future work
1.	Predicting Hit Songs with Machine Learning in year 2018	Logistic Regression, KNN, Gaussian Naïve Bayes, SVM	Bollywood Songs from Spotify api dataset.	Naïve Bayes – 60% LR, SVM, KNN – 50%	Choosing more futures like artist, artist popularity, label, genre and Feature selection would help to improve the accuracy rates.
2.	ANALYZING AND PREDICTING SONGS USING MACHINE LEARNING TECHNIQUES	Random forest, decision tree, logistic regression	Data from Kaggle	Random forest with more accuracy(79.0)	This work can be useful to filter a large music library for particular user's like and dislike. A music library filtered in this manner could further be used as input as user's interest.
3.	Predicting a Hit Song with Machine Learning	Logistic regression, decision tree and random forest, naïve bayes	Web API's Online Databases Global Music Charts	Naïve bayes – 51.1	Evaluating the music features by using algorithms can be a future endeavor. This would ensure the consistency of data

			Compiling Dataset		and particular features to be extracted that may prove to be more significant than the features extracted from Spotify's API. An in-depth breakdown of the lyrical sentiment is something that could enhance future work. Breaking down an amalgamation of slang words, onomatopoeia, and deeper meanings to use in future models will be worked on.
4.	Music Trend Prediction Based on Improved LSTM and Random Forest Algorithm	Improved LSTM and Random Forest Algorithm	W.E.K.A dataset	LSTM and SVM are reduced from the original 0.08 and 0.067 to 0.048 and 0.035	the following types of work can try to combine the prediction model to improve the prediction effect.
5.	Artificial Intelligence Induced Music Genre Prediction using KNN Architecture	Machine learning algorithms used is deep neural network,recurrent neural network, support vector machine, K nearest	The GTZAN dataset contains 1000 music files in .au format divided into ten genres,	Svm -74%	In future, model learns the data distribution of time series with long-term temporal relationships, as seen in actual data. Along with the audio Mel spectrogram, the suggested model utilised different

		neighbour, cnn	with 100 tracks in each category.		inputs for various models.
6.	Music Popularity Prediction Through Data Analysis of Music's Characteristics	Linear Regression,k nearest neighbour,random forest	The section analyzes the two different sets of data: Top 10 songs in 2010-2019 and Top 50 songs in 2019.	Linear regression with low rsme value – 3.12	expanding this project that can predict the popularity of music by analyzing the audio of music instead of text data.
7.	Automatically Predicting Popularity of Music Tracks Based on Lyrics	Knn,svm,naïve bayes	A dataset is built by combining data from the LFM-1b dataset	Svm -62%	this research could be improved and adapted to deliver better results in future studies. By devoting more attention on creating a balanced dataset, better and unbiased models could be trained. Such a dataset could be developed by adapting the dataset used for this research by undersampling the majority classes or oversampling the minority classes

8.	Music Genre Classification using Machine Learning Techniques	Methodolog y used here is data pre processing along with Using deep learning, we can achieve the task of music genre classification without the need for hand-crafted features. Convolution al neural net- works (CNNs) have been widely used for the task of image classification	we make use of Audio Set, which is a large-scale human annotate d database of sounds (Gemme ke et al., 2017).	the best performance in terms of all metrics is observed for the convolutional neural network model based on VGG-16 that uses only the spec- trogram to predict the music genre.	Futures studies can identify ways to pre-process this noisy data before feeding it into a machine learning model, in order to achieve better perfor- mance.
9.	CLASSIFICATI ON OF MUSICAL GENRE: A MACHINE LEARNING APPROACH	All the experiments have been run within the Waikato Environment for Knowledge Analysis (WEKA, ref. [6]).Various learning algorithms have been considered for ourexperime nts,	WEKA	Jazz and Blues classifiers are often misleading each other:when Jazz has low precision, the recall related to Blues goes down.	medium term target is also the realization of sensibly larger musical corpora, with different dimensions, class granularity and coverage

		including decision-tree, Bayesian and rulebased classifiers			
10.	Musical Predictions With an Embodied Interface for Musical Machine Learning year 2020	recurrent neural network	EMPI's MDRNN	the survey questions were analyzed with an aligned rank transform (ART) and two-way mixed-effects ANOVA procedure.	This research has demonstrated that EMPI can produce compelling music experiences within a lab setting. EMPI, and future embodied predictive instruments, hold substantial potential for enhancing and enabling musical creativity.
11.	Deep Learning in Music analysis Systems and published year 2019	deep learning algorithm	AotM-2011 dataset	DBN-based approach achieves 0.323 in warm-start and 0.478 in cold-start.RMSE of 0.325 in warm-start and 0.495 in cold-start.	the urgent need for multimodal datasets for MRS as well as for establishing a common agreement on evaluation metrics to be used for common tasks in MRS, for instance, APC.
12.	Applications of Machine Learning to Music Research: Phenomenon of	Integration of symbolic learning and numeric le	dataset from kaggle	naive approach 52.19 % approach 1	Future work in this pro ject will concentrate primarily on aspects of domain modelling.

	Musical Expression			(qual. domain theory) 57.10 % approach 2 (abstraction) 66.67 %	
13.	Deep Learning in Music analysis	Long Short Term Memory (LSTM) recurrent neural network (RNN) architecture	Muse-All Muse-Truncated	Sample 1 10 second clip of the ”Bach Midi” model Sample 2 16 second clip of the 7 RNN-NADE sequence Sample 3 11 second clip of the ”Piano roll” model trained on the ”Muse-All” dataset	Further work, that more work could be done in developing a better evaluation metric of the quality of a piece – only then will be able to train models that are truly able to compose original music
14.	Music Genre Classification Using CNN published year 2019	CNN Convolutional Neural Networks	GTZAN dataset	the best accuracy is 92.456564	In further this project can be done by various algorithms using deep learning algorithms and machine learning algorithm
15.	Music analysis of Classification using Deep learning techniques year published 2021	After data is processed and create first deep learning model. construct a Convolution	GITZAN dataset, which contains 1000 music files	for cnn the accuracy is 0.52 f1 score and for Transfer learning based model	in further it can be performed The transfer learning-based model has performed best among all three models. We have used the Keras

		Neural Network model with required input and out units		is 0.52 f1 score Multimodal-based model is 0.38 f1 score	framework for the implementation on the google Collaboratory platform.
16.	Music rhythm tree based partitioning approach to decision tree classifier published year 2019	Random Forest, Bagging, AdaBoost, an ensemble technique and a vertical partitioning method with respect to classification accuracy, standard deviation, misclassification rate and F-score	High dimensional datasets to evaluate superiority of the proposed method	MRTPDT + SMOTE + C5.0 show highest F-score for 20 blocks, 45 blocks and 15 blocks	In future, the project explore evolutionary techniques to do further improvement in the performance of MRTPDT method.
17.	Audio content analysis for online audio visual data segmentation and classification published year 2018	The multilayer neural network (MNN) and the hidden Markov model (HMM)	GITZAN dataset, which contains 10000 music files	from AR model parame- ters of order 40, and is calculated once every 400 input sam- ples. Each signal frame for computing the spectrum contains 512 samples	in future Methods are alsopro- posed for estimating the fundamental frequency and extracting spectral peak tracks from the AR model generated spectrum. Based on audio feature analysis

18.	Analyzing Music Using Neural Network	Nerual networks: CNN	Not mentioned.	CNN – 82%	Sense improvements should taken to get higher accuracy rate.
19.	Predicting the Music Mood of a Song with Deep Learning.	K-Fold Cross Validation	Spotify audio data	Accuracy was 72.75%.	This function takes the Id of the song as an argument and includes inside the Neural Network model created.
20.	ANALYZING AND PREDICTING SONGS USING MACHINE LEARNING TECHNIQUES in 2020	Random Forest, Decision tree, Logistic Regression.	Spotify Dataset	They conclude that Random forest shows higher accuracy rate.	We can work on some other different algorithms and follow the data mining techniques to improve the accuracy.
21.	Predicting Hit Songs with Machine Learning in year 2018	Logistic Regression, KNN, Gaussian Naïve Bayes, SVM	Bollywood Songs from Spotify api dataset.	Naïve Bayes – 60% LR, SVM, KNN – 50%	Choosing more futures like artist, artist popularity, label, genre and Feature selection would help to improve the accuracy rates.
22.	Mood based Music Recommendation System	Open cv, image classification, CNN.	Live feed using open cv.	Accuracy – 75%	Low accuracy.
23.	Music Service Data Analysis with Spark in year 2019	Random Forest Classifier;	Sparkify events dataset	Random Forest Classifier- 81%;	Large data set can give the robustness of the model.

		Logistic Regression Classifier; Gradient-Boosted Tree Classifier; Naïve Bayes Classifier.		Logistic Regression Classifier – 79%; Gradient-Boosted Tree Classifier – 75%; Naïve Bayes Classifier – 79%.	
24.	Experimenting with Music Taste Prediction by User Profiling.	KNN	Dataset obtained from smart radio	KNN – 55%	Need to work on data to improve the accuracy of prediction model and we can also use some other models to get higher accuracy by comparing them.
25.	HITPREDICT: PREDICTING HIT SONGS USING SPOTIFY DATA	Logistic Regression (LR), Gaussian Discriminant Analysis (GDA), Support Vector Machines (SVM), Decision Trees (DT), and Neural Networks (NN)	Spotify dataset	LR– 75%, GDA – 73%, SVM – 68%, DT -68%, and NN – 76%.	Observed that low accuracy rate and still improvements required.

26.	A music recommendation algorithm based on clustering and latent factor modein year 2020	K – means	Million Songs Dataset From National Science Foundation (IIS)	Instead of providing accuracy report they given the comparisons.	focus on the combination of latent factor models and other machine learning algorithms.
27.	Song Hit Prediction: Predicting Billboard Hits Using Spotify Data.	Logistic Regression (LR), Neural Network (NN), Random Forest (RF), Support Vector Machine (SVM)	Spotify data set	LR – 80%, NN – 83%, RF – 88%, SVM – 82%	More features such as artist info, song popularity will also help to increase the accuracy of the model.
28.	A Model for Predicting Music Popularity on Streaming Platforms in year 2020	SVM, GNB, LR, KNN.	Data collected using spotify api	SVM – 85%, GNB – 845, LR – 84%, KNN – 83%.	Still false prediction labels are there , consideration of artist, recording partner popularity helps to improve the model.
29.	Automatic Music Mood Detection Using Transfer Learning and Multilayer Perceptron	CNN	Million Song dataset	CNN – 72%–80%	Low accuracy since in many time CNN provide higher accuracy rate.

30.	Churn Analysis in a Music Streaming Service in year 2017	LR, RF, ANN, NAT.	The data consisted of 1.4 million users, sampled evenly over a period of 14 days. That's 50,000 users for each of the two classes, per day	F1 SCORE OF LR – 50%, RF -83%, ANN – 84%, NAT – 83%.	Improvising the techniques will helps to prediction model to produce more accurate results.
31.	Experience in Applying Data Mining Techniques to Musical Content Database to Identify Personality Traits	Random Forest, NN, Decision tree, naïve bayes.	Million Song dataset	First, the variables of Personality Inventory NEO-PI-R (see section 4.2) are measured, and then, through tests of correlational hypothesis and application of statistical techniques such as SEMMA Methodology	The influence of music on the behavior and psycho-emotional construction of people during their life at the personal, social, and cultural levels is undeniable. This paper has identified a mechanism capable of evaluating personality traits through five major factors centered on the OCEAN model

				the existing correlation is estimated	and different techniques of data mining to find the associations between musical preferences and personality traits.
32.	CLASSIFICATION OF MUSICAL GENRE: A MACHINE LEARNING APPROACH	Naïve bayes,VFI, J48.	Million Song dataset	Jazz and Blues classifiers are often misleading each other: when Jazz has low precision, the recall related to Blues goes down. This reflects the fact that in a multiclass categorizer a class, though being easily recognizable by itself, is shaded by the similar characteristics of more prominent classes	medium term target is also the realization of sensibly larger musical corpora, with different dimensions, class granularity and coverage

1.

Exploring the possibility of predicting hit songs is both interesting from a scientific point of view and something that could be beneficial to the music industry. In this research we raise the question if it is possible to classify a music track as a hit or a non-hit based on its audio features. We investigated which machine learning algorithms could be suited for a task like this. Four different models were built using various algorithms such as Support Vector Machine and Gaussian Naive Bayes. The obtained results do not indicate that it is possible to predict hit songs on our particular dataset. This stands in contrast to some previous research within this field. We discuss the potential problem in using only audio features, and how this seems not to be sufficient information for predicting a hit.

2.

Analysing and predicting songs whether a user like/dislike a song through exploratory data analysis and machine learning. The user data of custom like and dislike dataset is considered for the study. We apply machine learning algorithm to predict the song, whether user like or dislike it. The machine algorithm used for predictions are Logistics regression, Decision Tree and Random forest. We evaluate the three machine learning algorithm and through experimental analysis, we find the best algorithm for song like prediction. The problem of building a machine learning model which classifies music into liked or disliked according to user playlist. Another problem to be studied is to compare the accuracies of this machine learning model and the pre-existing models, and draw the necessary conclusions. Musical genre classification of audio signals, initially explored about how the automatic classification of audio signals into a hierarchy of musical genres is to be done.

3.

Thought to be an ever-changing art form, music has been a form of recreational entertainment for ages. The music industry is constantly making efforts for songs to be a hit and earn considerable revenues. It could be an interesting exercise to predict a song making it to top charts from a mathematical perspective. While several studies have looked into factors after a song is released, this research looks at apriori parameters of a song to predict the success of a song. Data sources available from multiple platforms are combined to create a dataset that has technical parameters of a song and sentimental analysis of the lyrics. Four machine learning algorithms (Logistic Regression, Decision Trees, Naïve Bayes and Random Forests) to answer the question-Is there a magical formula for the prediction of hit songs? It was found that there are elements beyond technical data points that could predict a song being hit or not. This paper takes a stand that music prediction is yet not a data science activity.

4.

Music has the ability to evoke different emotions in people, which is reflected in their physiological signals. Advances in affective computing have introduced computational methods to analyse these signals and understand the relationship between music and emotion in greater detail. We analyse Electrodermal Activity (EDA), Blood Volume Pulse (BVP), Skin Temperature (ST) and Pupil Dilation (PD) collected from 24 participants while they listen to 12 pieces from 3 different genres of music. A set of 34 features were extracted from each signal and 6 different feature selection methods were applied to identify useful features. Empirical analysis shows that a neural network (NN) with a set of features extracted from the physiological signals can achieve 99.2% accuracy in differentiating among the 3 music genres.

5.

The exponential growth of online music streaming has given birth to many new platforms among which, the widely used platform is Spotify. The most popular music streaming app's data can be used to predict the capability of a song to be popular before its release with the help of attributes like loudness, energy, acousticness, etc. which is defined when thesong is being made. This study helps to predict the popularity of the song using the song metrics available in Spotify by applying Random Forest classifier, K-Nearest neighbour classifier and Linear Support Vector classifier to compare which of these models can effectively predict the popularity. The results found that Random Forest works the best for predicting popularity with high accuracy, precision, recall and F1-score.

6.

Predicting song popularity is particularly important in keeping businesses competitive within a growing music industry. But what exactly makes a song popular? Starting with the Million Song Dataset, a collection of audio features and metadata for approximately one million songs, we evaluated different classification and regression algorithms on their ability to predict popularity and determined the types of features that hold the most predictive power.

7.

Categorizing music files according to their genre is a challenging task in the area of music information retrieval (MIR). In this study, we compare the performance of two classes of models. The first is a deep learning approach wherein a CNN model is trained end-to-end, to predict the genre label of an audio signal, solely using its spectrogram. The second approach utilizes hand-crafted features, both from the time domain and frequency domain. We train four traditional machine learning classifiers with these features and compare their performance. The features that contribute the most towards this classification task are identified. The experiments are conducted on the Audio set data set and we report an AUC value of 0.894 for an ensemble classifier which combines the two proposed approaches.

8.

In this paper, we investigate the impact of machine learning algorithms in the development of automatic music classification models aiming to capture genres distinctions. The study of genres as bodies of musical items aggregated according to subjective and local criteria requires corresponding inductive models of such a notion. This process can be thus modeled as an example-driven learning task. We investigated the impact of different musical features on the inductive accuracy by first creating a medium-sized collection of examples for widely recognized genres and then evaluating the performances of different learning algorithms. In this work, features are derived from the MIDI transcriptions of the song collection.

9.

Machine-learning models of music often exist outside the worlds of musical performance practice and abstracted from the physical gestures of musicians. In this work, we consider how a recurrent neural network (RNN) model of simple music gestures may be integrated into a physical instrument so that predictions are sonically and physically entwined with the performer's actions.

10.

This chapter describes an application of machine learning techniques to the study of a fundamental phenomenon in tonal music. Learning algorithms are described that induce general rules of expressive music performance from example of real performances by musicians. Motivated by the insight that general knowledge about music plays an essential role in the way humans learn this task, we present two alternative approaches to knowledge-based learning. In both cases, the domain knowledge provided to the learner is based on established theories of tonal music. Experimental results show that both approaches lead to a significant improvement of the learning results, compared to purely inductive learning.

11.

In addition to traditional tasks such as prediction, classification and translation, deep learning is receiving growing attention as an approach for music generation, as witnessed by recent research groups such as Magenta at Google and CTRL (Creator Technology Research Lab) at Spotify. The motivation is in using the capacity of deep learning architectures and training techniques to automatically learn musical styles from arbitrary musical corpora and then to generate samples from the estimated distribution. However, a direct application of deep learning to generate content rapidly reaches limits as the generated content tends to mimic the training set without exhibiting true creativity. Moreover, deep learning architectures do not offer direct ways for controlling generation (e.g., imposing some tonality or other arbitrary constraints). Furthermore, deep learning architectures alone are autistic automata which generate music autonomously without human user interaction, far from the objective of interactively assisting musicians to compose and refine music.

12.

Music plays a very important role in people’s lives. Music bring like-minded people together and is the glue that holds communities together. Communities can be recognized by the type of songs that they compose, or even listen to. The purpose of our project and research is to find a better machine learning algorithm than the pre-existing models that predicts the genre of songs. In this project, we built multiple classification models and trained them over the Free Music Archive (FMA) dataset. We have compared the performances of all these models and logged their results in terms of prediction accuracies. Few of these models are trained on the mel-spectrograms of the songs along with their audio features, and few others are trained solely on the spectrograms of the songs. It is found that one of the models, a convolutional neural network, which was given just the spectrograms as the dataset, has given the highest accuracy amongst all other models.

13.

This analyse Electrodermal Activity (EDA), Blood Volume Pulse (BVP), Skin Temperature (ST) and Pupil Dilation (PD) collected from 24 participants while they listen to 12 pieces from 3 different genres of music. A set of 34 features were extracted from each signal and 6 different feature selection methods were applied to identify useful features. Empirical analysis shows that a neural network (NN) with a set of features extracted from the physiological signals can achieve 99.2% accuracy in differentiating among the 3 music genres. The model also reaches 98.5% accuracy in classification based on participants’ subjective rating of emotion. The paper also identifies some useful features to improve accuracy of the classification models.

14.

Decision tree is a widely used non-parametric technique in machine learning, data mining and pattern recognition. It is simple to understand and interpret, however it faces challenges such as handling higher dimensional and class imbalanced datasets, over-fitting and instability. To overcome some of these issues, vertical partitioning approaches like serial partitioning, theme based partitioning are used in the literature. A vertical partitioning approach divides the feature set into subsets of features (blocks) and makes use of these subsets for subsequent tasks. In this work, we use the ideas of music rhythm tree to propose a novel vertical partitioning technique. It orders the features based on the average correlation strength of the features before partitioning the feature set. The proposed method is proved to be superior by showing an average of 13.8%,6%,9.8%,19.7%,9.4%, and 29.4% higher classification accuracy over C4.5, Random Forest, Bagging, Adaboost, an ensemble technique and a vertical partitioning technique respectively. Our empirical results on 15 datasets demonstrate that the proposed vertical partitioning method is more stable and better in handling class-imbalanced data. Finally, some popular statistical tests are conducted to validate the statistical significance of the results of the proposed method.

15.

While current approaches for audiovisual data segmentation and classification are mostly focused on visual cues, audio signals may actually play a more important role in content parsing for many applications. An approach to automatic segmentation and classification of audiovisual data based on audio content analysis is proposed. The audio signal from movies or TV programs is segmented and classified into basic types such as speech, music, song, environmental sound, speech with music background, environmental sound with music background, silence, etc. Simple audio features including the energy function, the average zero-crossing rate, the fundamental frequency, and the spectral peak tracks are extracted to ensure the feasibility of real-time processing. A heuristic rule-based procedure is proposed to segment and classify audio signals and built upon morphological and statistical analysis of the time-varying functions of these audio features. Experimental results show that the proposed scheme achieves an accuracy rate of more than 90% in audio classification.

16.

The focus of this work is to study how to efficiently tailor Convolutional Neural Networks (CNNs) towards learning timbre representations from log-mel magnitude spectrograms. We first review the trends when designing CNN architectures. Through this literature overview we discuss which are the crucial points to consider for efficiently learning timbre representations using CNNs. From this discussion we propose a design strategy meant to capture the relevant time-frequency contexts for learning timbre, which permits using domain knowledge for designing architectures. In addition, one of our main goals is to design efficient CNN architectures – what reduces the risk of these models to over-fit, since CNNs' number of parameters is minimized. Several architectures based on the design principles we propose are successfully assessed for different research tasks related to timbre: singing voice phoneme classification, musical instrument recognition and music auto-tagging.

17.

This paper proposes a novel machine learning approach for the task of on-line continuous-time music mood regression, i.e., low latency prediction of the time-varying arousal and valence in musical pieces. On the front-end, a large set of segmental acoustic features is extracted to model short-term variations. Then, multi-variate regression is performed by deep recurrent neural networks to model longer-range context and capture the time-varying emotional profile of musical pieces appropriately. Evaluation is done on the 2013 Media Eval Challenge corpus consisting of 1 000 pieces annotated in continuous time and continuous arousal and valence by crowdsourcing. In the result, recurrent neural networks outperform SVR and feedforward neural networks both in continuous-time and static music mood regression, and achieve an R 2 of up to .70 and .50 with arousal and valence annotations.

18.

The focus of a survey paper on music classification and tagging in 2011 revealed the previous trends in the field. Most of the 149 papers surveyed therein were based on the "conventional" machine-learning framework, which involves a pipeline of feature extraction and classifier learning. The features were mostly manually designed to succinctly represent acoustic or musical characteristics given the task.

19.

Music recommender systems (MRSs) have experienced a boom in recent years, thanks to the emergence and success of online streaming services, which nowadays make available almost all music in the world at the user's fingertip. While today's MRSs considerably help users to find interesting music in these huge catalogs, MRS research is still facing substantial challenges. In particular when it comes to build, incorporate, and evaluate recommendation strategies that integrate information beyond simple user–item interactions or content-based descriptors, but dig deep into the very essence of listener needs, preferences, and intentions, MRS research becomes a big endeavor and related publications quite sparse. The purpose of this trends and survey article is twofold.

20.

A recommendation system is a program that utilizes techniques to suggest to a user items that they would likely prefer. This paper focuses on an approach to improving music recommendation systems, although the proposed solution could be applied to many different platforms and domains, including Youtube (videos), Netflix (movies), Amazon (shopping), etc. Current systems lack adequate efficiency once more variables are introduced. In this paper algorithm, Tunes Recommendation System (T-RECSYS), uses a hybrid of content-based and collaborative filtering as input to a deep learning classification model to produce an accurate recommendation system with real-time prediction. We apply our approach to data obtained from the Spotify Recsys Challenge, attaining precision scores as high as 88% at a balanced discrimination threshold.

21.

With commercial music streaming service which can be accessed from mobile devices, the availability of digital music currently is abundant compared to previous era. Sorting out all this digital music is a very time-consuming and causes information fatigue. Therefore, it is very useful to develop a music recommender system that can search in the music libraries automatically and suggest suitable songs to users. By using music recommender system, the music provider can predict and then offer the appropriate songs to their users based on the characteristics of the music that has been heard previously. Our research would like to develop a music recommender system that can give recommendations based on similarity of features on audio signal. This study uses convolutional recurrent neural network (CRNN) for feature extraction and similarity distance to look similarity between features. The results of this study indicate that users prefer recommendations that consider music genres compared to recommendations based solely on similarity.

22.

In the current study, they approached the Hit Song Science problem, aiming to predict which songs will become Billboard Hot 100 hits. We collated a dataset of approximately 4,000 hit and non-hit songs and extracted each songs audio features from the Spotify Web API. We were able to predict the Billboard success of a song with approximately 75% accuracy on the validation set, using five machine-learning algorithms. The most successful algorithms were Logistic Regression and a Neural Network with one hidden layer.

23.

Recommender systems have been widely used in various domains including movies, news, music with an aim to provide the most relevant proposals to users from a variety of available options. Recommender systems are designed using techniques from many fields, some of which are: machine learning, information retrieval, data mining, linear algebra and artificial intelligence. Though in-memory nearest-neighbour computation is a typical approach for collaborative filtering due to its high recommendation accuracy.

24.

In this work, they attempt to solve the Hit Song Science problem, which aims to predict which songs will become chart-topping hits. We constructed a dataset with approximately 1.8 million hit and non-hit songs and extracted their audio features using the Spotify Web API. We test four models on our dataset. Our best model was random forest, which was able to predict Billboard song success with 88% accuracy.

25.

The continuous evolution of multimedia applications is fostering applied research in order to dynamically enhance the services provided by platforms such as Spotify, Lastfm, or Billboard. Thus, innovative methods for retrieving specific information from large volumes of data related with music arises as a potential challenge within the Music Information Retrieval (MIR) framework. Moreover, despite the existence of several musical-based datasets, there is still a lack of information to properly assess an accurate estimation of the impact or the popularity of a song within a platform. Furthermore, the aforementioned platforms measure the popularity in various manners, thus increasing the difficulties in performing generalized and comparable models. In this paper, the creation of SpotGenTrack Popularity Dataset (SPD) is presented as an alternative solution to existing datasets that will facilitate researchers when comparing and promoting their models. In addition, an innovative multimodal end-to-end Deep Learning architecture named as HitMusicNet is presented for predicting popularity in music recordings. Experiments conducted show that the proposed architecture outperforms previous studies in the State-of-the-Art by incorporating three main modalities to the analysis, such as audio, lyrics and meta-data as well as a preliminary compression stage via autoencoder to better the capability of the model when predicting the popularity.

26.

This paper proposes an automatic mood detection of music with a composition of transfer learning and multilayer. The five layered convolutional neural network pre-trained on Million Song dataset is used to extract the features from EmoMusic dataset. We obtain a set of features from the different five layers, which is fed into multilayer perceptron (MLP)-based regression. Through the regression network we estimate the mood of music on Thayer's two-dimensional emotion space, which consists of the axes corresponding to arousal and valence. Because the EmoMusic dataset does not provide enough number of data for training, we augment the data by time stretching to make it tripled. We perform the experiment with the augmented data as well as the original EmoMusic dataset. Box and whisker plot along with the mean of 10-fold cross-validation has been used for evaluating the proposed mood detection. In terms of the percentage of R^2 score for measure of accuracy, the proposed MLP shows state-of-the-art estimates for the augmented EmoMusic dataset.

27.

Service subscription, which is known as customer churn. For that goal, an explainable predictive customer churn model is an essential tool to be owned by a telecommunication provider. In this paper, we developed the explainable model by utilizing the concept of vector embedding in Deep Learning. We show that the model can reveal churning customers that can potentially be converted back to use the previous telecommunication service. The generated vectors are also highly discriminative between the churning and loyal customers, which enable the developed models to be highly predictive for determining whether a customer would cease his/her service subscription or not. The best model in our experiment achieved a predictive performance of 81.16%, measured by the F1

Score. Further analysis on the clusters similarity and t-SNE plot also confirmed that the generated vectors are discriminative for churn prediction.

28.

In present-day in consumers' homes, there are millions of Internet-connected devices that are known to jointly represent the Internet of Things (IoT). The development of the IoT industry has led to the emergence of connected devices and home assistants that create smart living environments. However, the continuously generated data accumulated by these connected devices create security issues and raise user's privacy concerns. The present study aims to explore the main security issues in smart living environments using data mining techniques. To this end, we applied a three-sentence data mining analysis of 938,258 tweets collected from Twitter under the user-generated data (UGD) framework. First, sentiment analysis was applied using Textblob which was tested with support vector classifier, multinomial naïve bayes, logistic regression, and random forest classifier; as a result, the analyzed tweets were divided into those expressing positive, negative, and neutral sentiment. Next, a Latent Dirichlet Allocation (LDA) algorithm was applied to divide the sample into topics related to security issues in smart living environments.

29.

Music genre prediction is one of the topics that digital music processing is interested in. In this study, acoustic features of music have been extracted by using digital signal processing techniques and then music genre classification and music recommendations have been made by using machine learning methods. In addition, convolutional neural networks, which are deep learning methods, were used for genre classification and music recommendation and performance comparison of the obtained results has been. In the study, GTZAN database has been used and the highest success was obtained with the SVM algorithm.

V. Existing Systems:

a) The existing systems have the accuracy of Naïve bayes 60%,SVM 68%,LR 72%,random forest 80% are reported.
b) There are no hybrid models conducted in the existing systems.

VI. Gap identified:

There is no good accuracy, so need to perform hybrid model use top 4 accuracy models using voting classifier.

VII.DATASETS DESCRIPTION & SAMPLE DATA

Data Set Information:

The data set was taken from the Kaggle. which comprised of 2017 songs from Spotify with 16 different attributes and an output as 1 if the audience might like that particular song and 0 if the audience would not like that particular song. We are going to use these data set as training and testing of our prediction model.

Link: https://www.kaggle.com/datasets/mrmorj/dataset-of-songs-in-spotify

Attribute Information:

The dataset consists of following columns :

Tempo: The tempo of the song. The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, the tempo is the speed or pace of a given piece and derives directly from the average beat duration.

Energy: Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. Higher the value more energetic the song.

Danceability: Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. The value ranges from 0 to 1. Higher the value more suitable the song is for dancing.

Loudness: Loudness values are averaged across the entire track. It is the quality of a song. It ranges from -60 to 0 DB. Higher the value, the louder the song.

Valence: A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

Liveness: Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides a strong likelihood that the track is live.

Acousticness: A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.

Speechiness: Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audiobook, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.

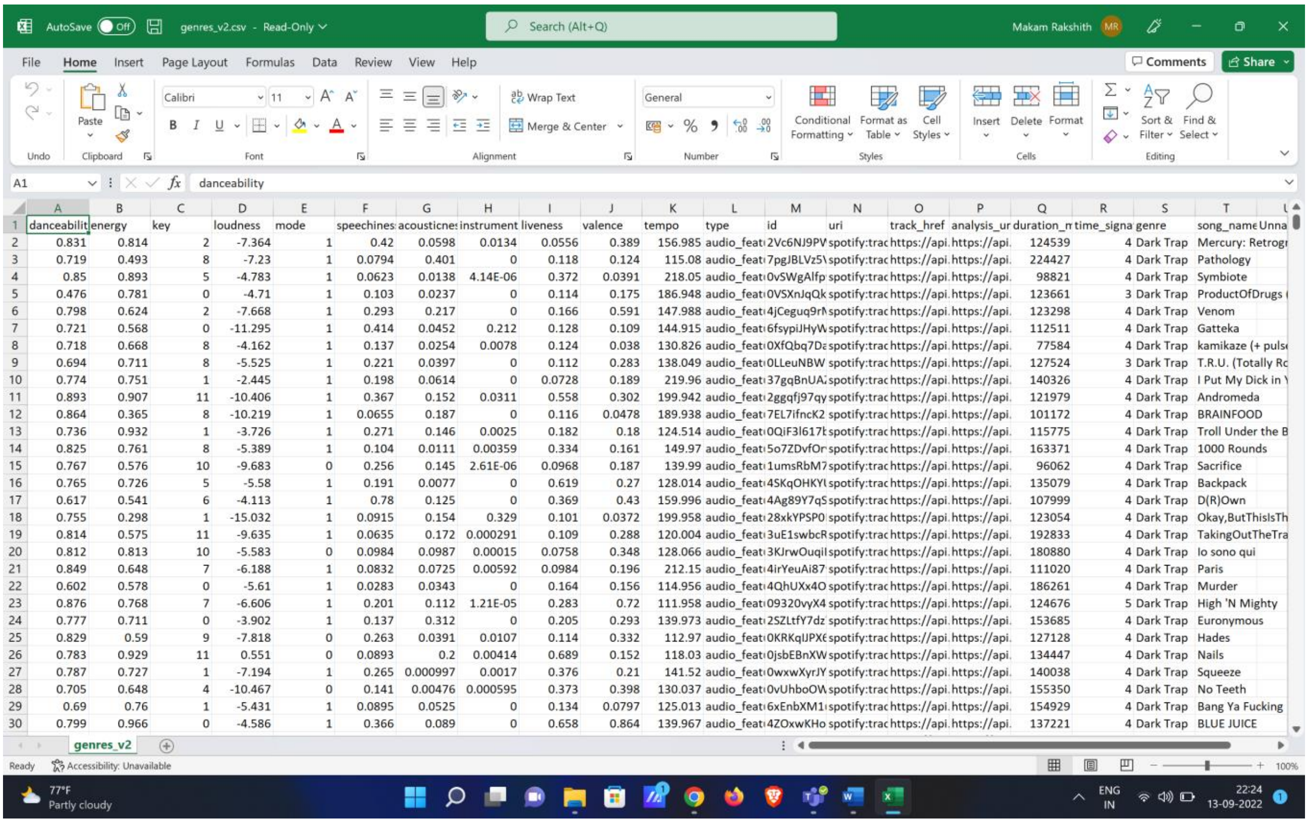
Mode: Songs can be classified as major and minor. 1.0 represents major mode and 0 represents minor.

Key: Key is the pitch, notes or scale of song that forms the basis of a song. 12 keys are ranging from 0 to 11.

Artists: Artist of the song

Target: 1 represent song is liked and 0 represent the song id disliked.

Sample Dataset:

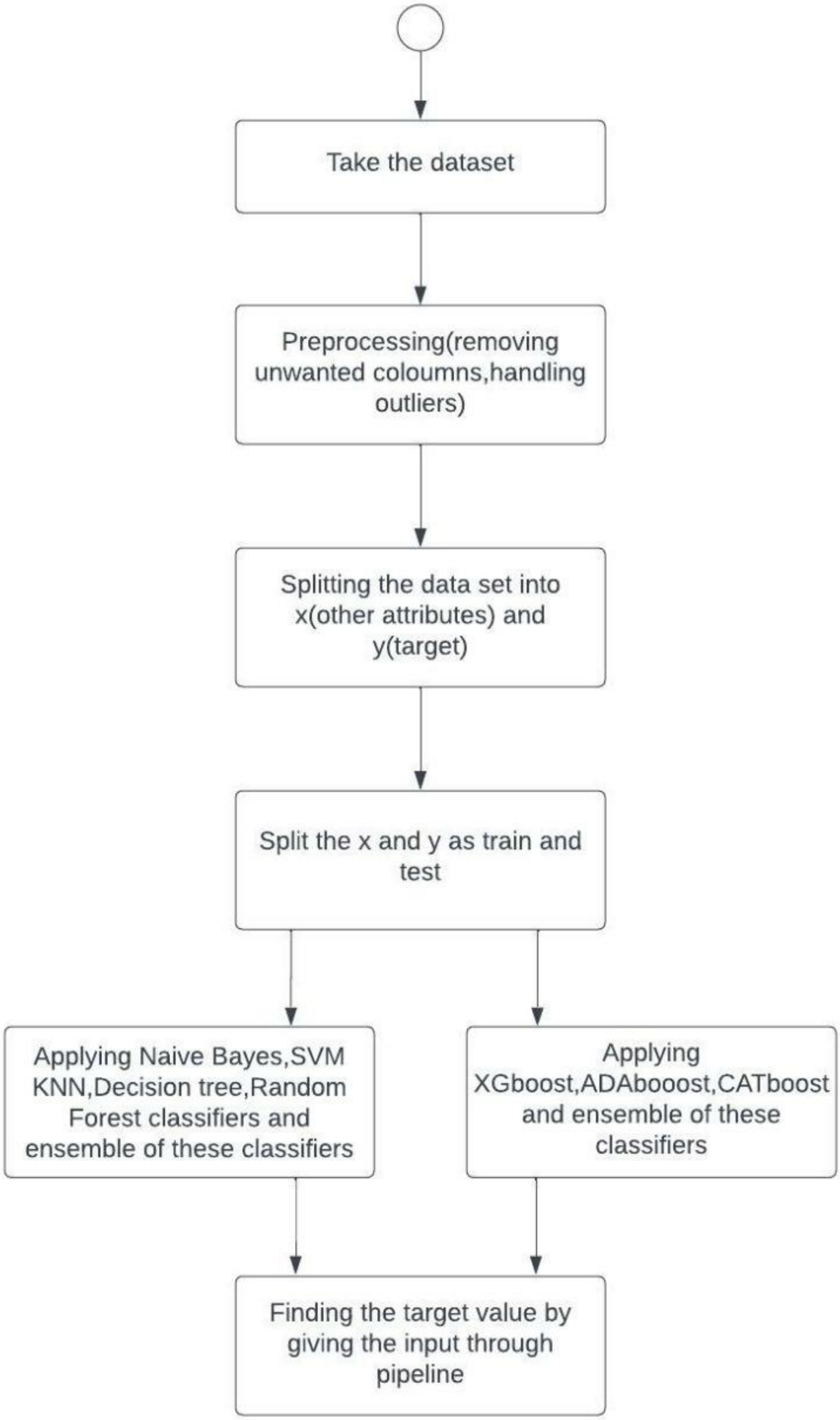


	danceabilit	energy	key	loudness	mode	speechines	acousticne	instrument	liveness	valence	tempo	type	id	uri	track_href	analysis_ur	duration_m	time_signa	genre	song_name	Unna
2	0.831	0.814	2	-7.364	1	0.42	0.0598	0.0134	0.0556	0.389	156.985	audio_feat	2Vc6NJ9PV	spotify:trac	https://api.	https://api.	124539	4	Dark Trap	Mercury: Retrogr	
3	0.719	0.493	8	-7.23	1	0.0794	0.401	0	0.118	0.124	115.08	audio_feat	7pgJBLVz5	spotify:trac	https://api.	https://api.	224427	4	Dark Trap	Pathology	
4	0.85	0.893	5	-4.783	1	0.0623	0.0138	4.14E-06	0.372	0.0391	218.05	audio_feat	0vSWgAlfp	spotify:trac	https://api.	https://api.	98821	4	Dark Trap	Symbiote	
5	0.476	0.781	0	-4.71	1	0.103	0.0237	0	0.114	0.175	186.948	audio_feat	0VSXnJqQk	spotify:trac	https://api.	https://api.	123661	3	Dark Trap	ProductOfDrugs (	
6	0.798	0.624	2	-7.668	1	0.293	0.217	0	0.166	0.591	147.988	audio_feat	4jCeguq9rN	spotify:trac	https://api.	https://api.	123298	4	Dark Trap	Venom	
7	0.721	0.568	0	-11.295	1	0.414	0.0452	0.212	0.128	0.109	144.915	audio_feat	6fsypiJHyW	spotify:trac	https://api.	https://api.	112511	4	Dark Trap	Gatteka	
8	0.718	0.668	8	-4.162	1	0.137	0.0254	0.0078	0.124	0.038	130.826	audio_feat	0XfQbq7Dz	spotify:trac	https://api.	https://api.	77584	4	Dark Trap	kamikaze (+ puls	
9	0.694	0.711	8	-5.525	1	0.221	0.0397	0	0.112	0.283	138.049	audio_feat	0LLeuNBW	spotify:trac	https://api.	https://api.	127524	3	Dark Trap	T.R.U. (Totally Rc	
10	0.774	0.751	1	-2.445	1	0.198	0.0614	0	0.0728	0.189	219.96	audio_feat	37gqBnUA	spotify:trac	https://api.	https://api.	140326	4	Dark Trap	I Put My Dick in Y	
11	0.893	0.907	11	-10.406	1	0.367	0.152	0.0311	0.558	0.302	199.942	audio_feat	2ggqfj97qy	spotify:trac	https://api.	https://api.	121979	4	Dark Trap	Andromeda	
12	0.864	0.365	8	-10.219	1	0.0655	0.187	0	0.116	0.0478	189.938	audio_feat	7EL7ifncK2	spotify:trac	https://api.	https://api.	101172	4	Dark Trap	BRAINFOOD	
13	0.736	0.932	1	-3.726	1	0.271	0.146	0.0025	0.182	0.18	124.514	audio_feat	0QiF3l617k	spotify:trac	https://api.	https://api.	115775	4	Dark Trap	Troll Under the B	
14	0.825	0.761	8	-5.389	1	0.104	0.0111	0.00359	0.334	0.161	149.97	audio_feat	5o7ZDvfOr	spotify:trac	https://api.	https://api.	163371	4	Dark Trap	1000 Rounds	
15	0.767	0.576	10	-9.683	0	0.256	0.145	2.61E-06	0.0968	0.187	139.99	audio_feat	1umsRbM7	spotify:trac	https://api.	https://api.	96062	4	Dark Trap	Sacrifice	
16	0.765	0.726	5	-5.58	1	0.191	0.0077	0	0.619	0.27	128.014	audio_feat	4SKqOHKY	spotify:trac	https://api.	https://api.	135079	4	Dark Trap	Backpack	
17	0.617	0.541	6	-4.113	1	0.78	0.125	0	0.369	0.43	159.996	audio_feat	4Ag89Y7qS	spotify:trac	https://api.	https://api.	107999	4	Dark Trap	D(R)Own	
18	0.755	0.298	1	-15.032	1	0.0915	0.154	0.329	0.101	0.0372	199.958	audio_feat	28xkYPSP0	spotify:trac	https://api.	https://api.	123054	4	Dark Trap	Okay,ButThisIsTh	
19	0.814	0.575	11	-9.635	1	0.0635	0.172	0.000291	0.109	0.288	120.004	audio_feat	3uE1swbcR	spotify:trac	https://api.	https://api.	192833	4	Dark Trap	TakingOutTheTra	
20	0.812	0.813	10	-5.583	0	0.0984	0.0987	0.00015	0.0758	0.348	128.066	audio_feat	3KJrwOuqil	spotify:trac	https://api.	https://api.	180880	4	Dark Trap	Io sono qui	
21	0.849	0.648	7	-6.188	1	0.0832	0.0725	0.00592	0.0984	0.196	212.15	audio_feat	4irYeuAi87	spotify:trac	https://api.	https://api.	111020	4	Dark Trap	Paris	
22	0.602	0.578	0	-5.61	1	0.0283	0.0343	0	0.164	0.156	114.956	audio_feat	4QhUXx4O	spotify:trac	https://api.	https://api.	186261	4	Dark Trap	Murder	
23	0.876	0.768	7	-6.606	1	0.201	0.112	1.21E-05	0.283	0.72	111.958	audio_feat	09320vyX4	spotify:trac	https://api.	https://api.	124676	5	Dark Trap	High 'N Mighty	
24	0.777	0.711	0	-3.902	1	0.137	0.312	0	0.205	0.293	139.973	audio_feat	2SZLtfY7dz	spotify:trac	https://api.	https://api.	153685	4	Dark Trap	Euronymous	
25	0.829	0.59	9	-7.818	0	0.263	0.0391	0.0107	0.114	0.332	112.97	audio_feat	0KRKqIJPX	spotify:trac	https://api.	https://api.	127128	4	Dark Trap	Hades	
26	0.783	0.929	11	0.551	0	0.0893	0.2	0.00414	0.689	0.152	118.03	audio_feat	0jsbEBnXW	spotify:trac	https://api.	https://api.	134447	4	Dark Trap	Nails	
27	0.787	0.727	1	-7.194	1	0.265	0.000997	0.0017	0.376	0.21	141.52	audio_feat	0wxwXyrJY	spotify:trac	https://api.	https://api.	140038	4	Dark Trap	Squeeze	
28	0.705	0.648	4	-10.467	0	0.141	0.00476	0.000595	0.373	0.398	130.037	audio_feat	0vUhboOW	spotify:trac	https://api.	https://api.	155350	4	Dark Trap	No Teeth	
29	0.69	0.76	1	-5.431	1	0.0895	0.0525	0	0.134	0.0797	125.013	audio_feat	6xEnbXM1	spotify:trac	https://api.	https://api.	154929	4	Dark Trap	Bang Ya Fucking	
30	0.799	0.966	0	-4.586	1	0.366	0.089	0	0.658	0.864	139.967	audio_feat	4ZOxwKHo	spotify:trac	https://api.	https://api.	137221	4	Dark Trap	BLUE JUICE	

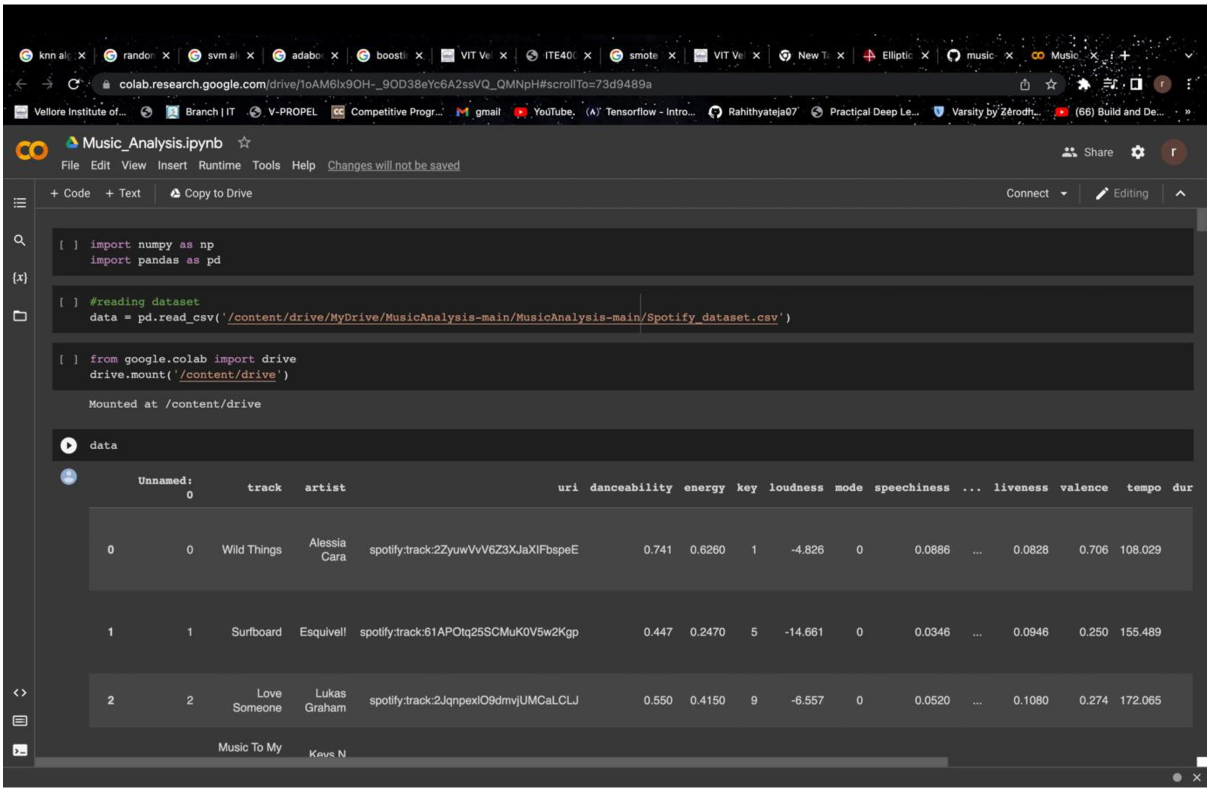
VIII.Proposed Model With Flowchart:

The algorithms we are going to use are decision tree classifier, Random forest classifier, XGB classifier, Naïve bayes, SVM, KNN, ADAboost clasdsifier, CATboost classifier.

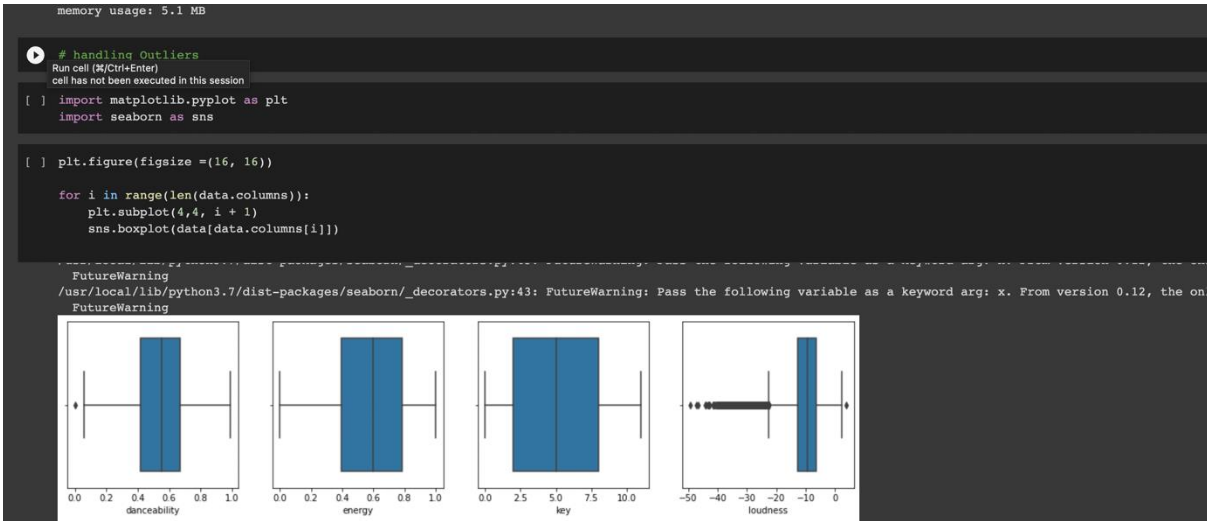
By taking best three classifier models individual accuracy we are going to made ensemble learning model and classify using voting classifier.


Take the dataset
Preprocessing(removing unwanted coloumns,handling outliers)
Splitting the data set into x(other attributes) and y(target)
Split the x and y as train and test
Applying Naive Bayes,SVM KNN,Decision tree,Random Forest classifiers and ensemble of these classifiers
Applying XGboost,ADAbooost,CATboost and ensemble of these classifiers
Finding the target value by giving the input through pipeline

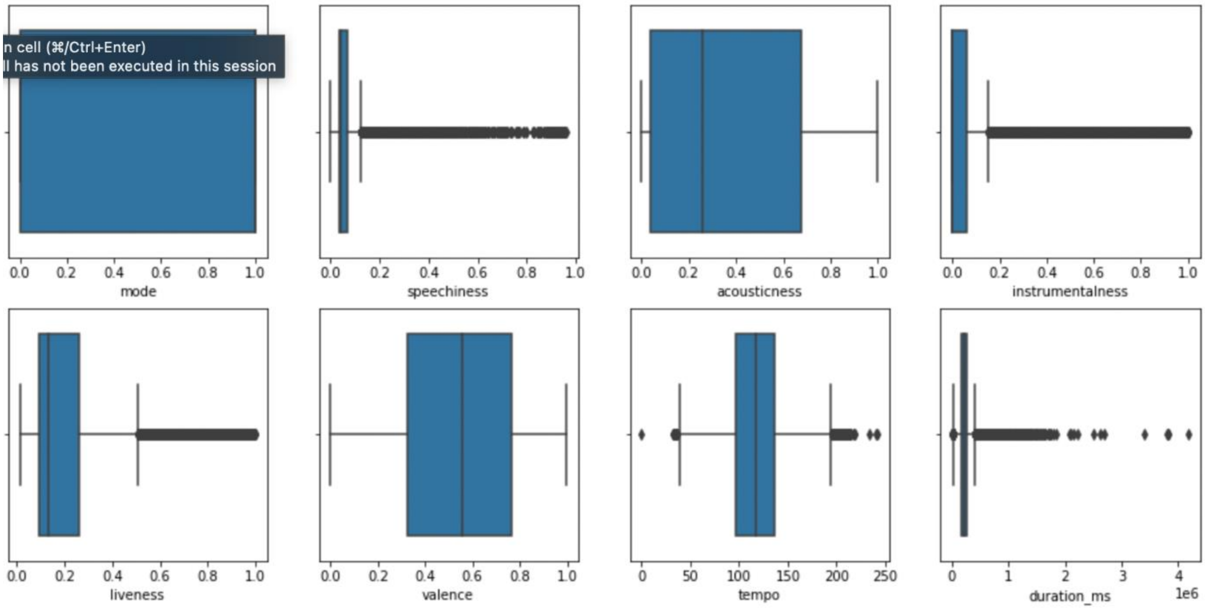
Our Results:

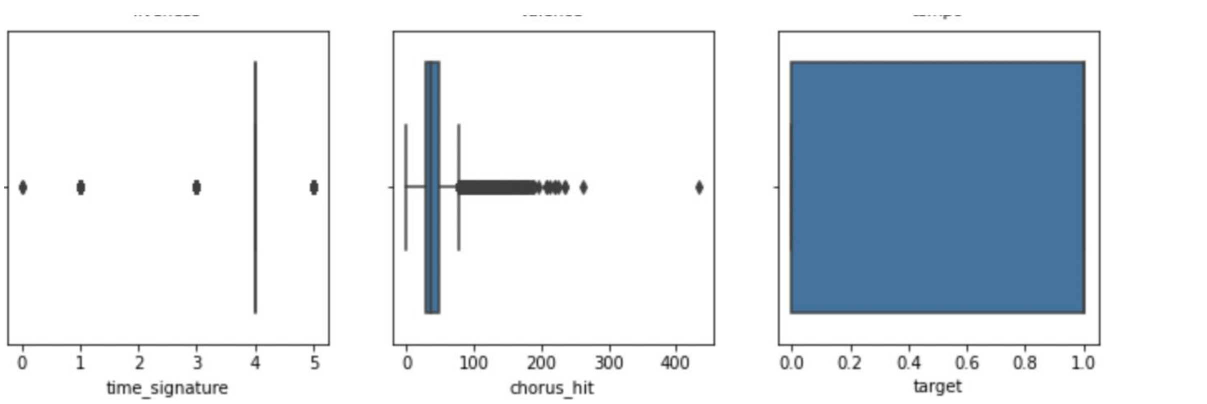


Music_Analysis.ipynb
import numpy as np
import pandas as pd
#reading dataset
data = pd.read_csv('/content/drive/MyDrive/MusicAnalysis-main/MusicAnalysis-main/Spotify_dataset.csv')
from google.colab import drive
drive.mount('/content/drive')
Mounted at /content/drive
data



memory usage: 5.1 MB
# handling Outliers
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize =(16, 16))
for i in range(len(data.columns)):
plt.subplot(4,4, i + 1)
sns.boxplot(data[data.columns[i]])
danceability
energy
key
loudness


n cell (⌘/Ctrl+Enter)
ll has not been executed in this session
mode
speechiness
acousticness
instrumentalness
liveness
valence
tempo
duration_ms
1e6


time_signature
chorus_hit
target

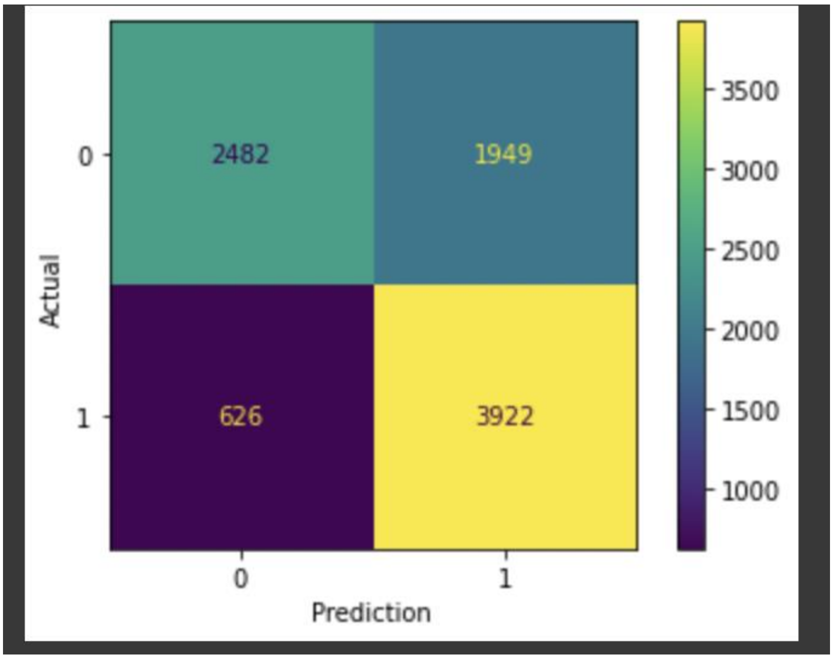
Naïve-bayes:

```
# NAIVE - BAYES
from sklearn.naive_bayes import GaussianNB
nb_model = GaussianNB()
nb_model.fit(x_train, y_train)
nb_pred = nb_model.predict(x_test)
acc_nb_model = metrics.accuracy_score(y_test,nb_pred)
print("accuracy: ", acc_nb_model)
```

```
accuracy:  0.7132197349370754
```

```
[ ] print(classification_report(y_test, nb_pred.round()))
```

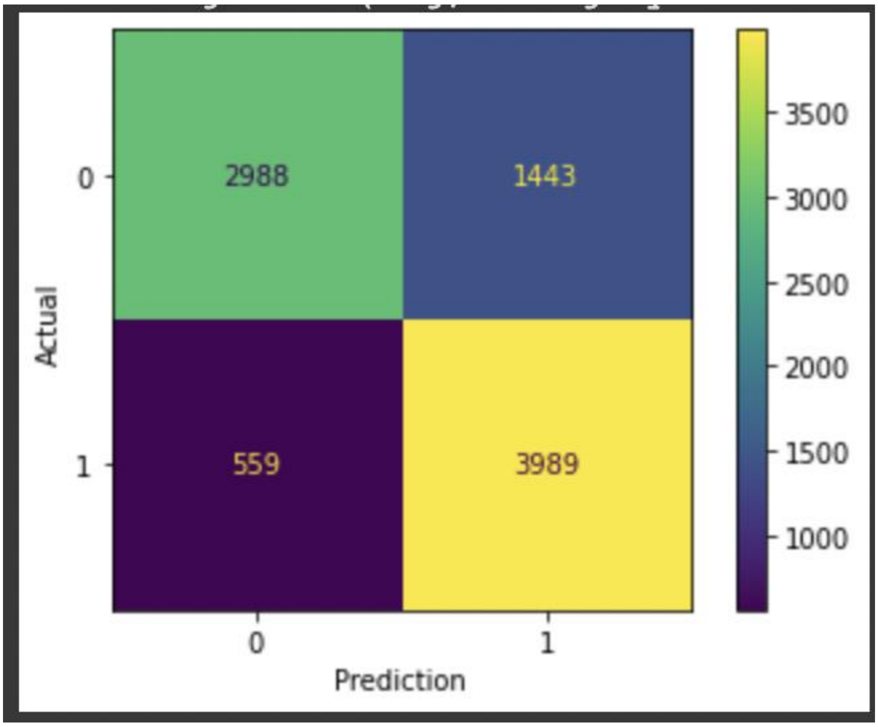
```
              precision    recall  f1-score   support

           0       0.80      0.56      0.66      4431
           1       0.67      0.86      0.75      4548

    accuracy                           0.71      8979
   macro avg       0.73      0.71      0.71      8979
weighted avg       0.73      0.71      0.71      8979
```



SVM(support vector machine):

```
[ ] #SVM
    from sklearn.svm import SVC
    svm_model = SVC()
    svm_model.fit(x_train, y_train)
    svm_pred = svm_model.predict(x_test)
    acc_svm_model = metrics.accuracy_score(y_test, svm_pred)
    print("accuracy: ", acc_svm_model)

    accuracy:  0.7770353045996213
```

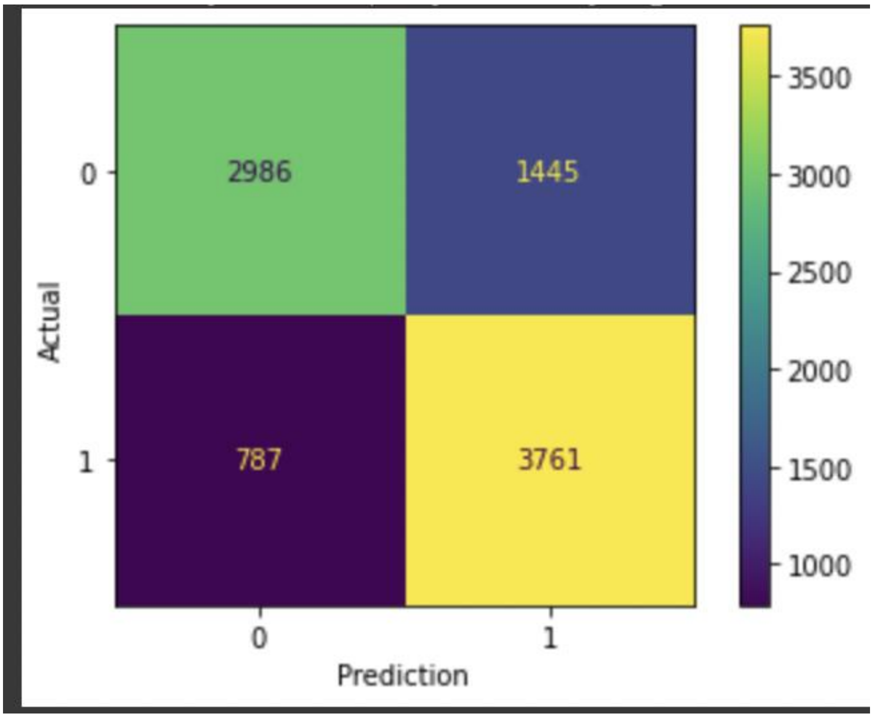
```
print(classification_report(y_test, svm_pred.round()))

              precision    recall  f1-score   support

           0       0.84      0.67      0.75      4431
           1       0.73      0.88      0.80      4548

    accuracy                           0.78      8979
   macro avg       0.79      0.78      0.77      8979
weighted avg       0.79      0.78      0.77      8979
```



KNN:

```
[ ] #KNN
    from sklearn.neighbors import KNeighborsClassifier
    KNN_model=KNeighborsClassifier()
    KNN_model.fit(x_train, y_train)
    KNN_pred = KNN_model.predict(x_test)
    acc_KNN_model = metrics.accuracy_score(y_test,KNN_pred)
    print("accuracy: ", acc_KNN_model)

    accuracy:  0.7514199799532242
```

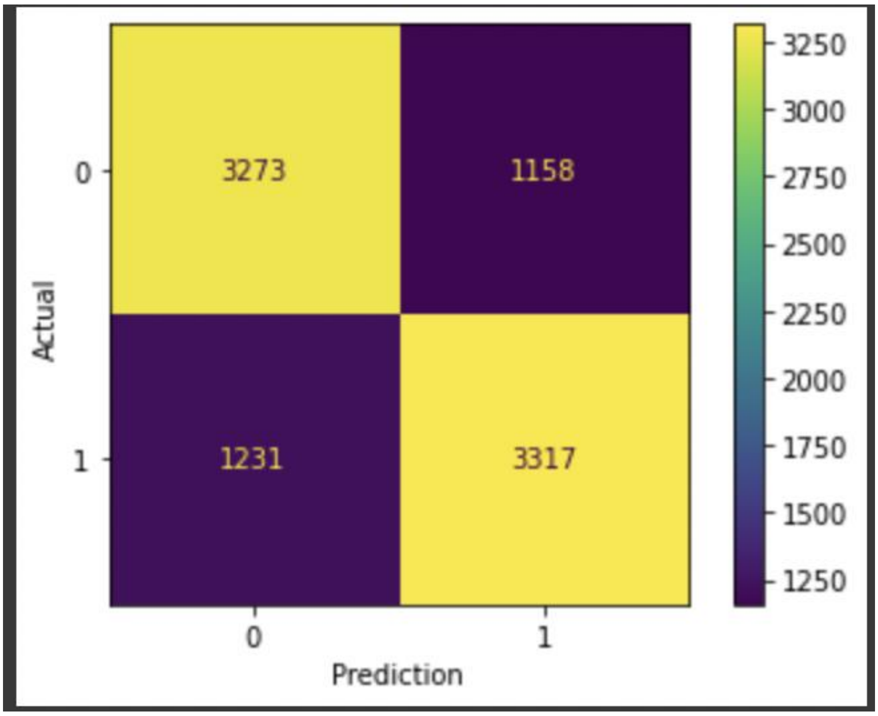
```
print(classification_report(y_test, KNN_pred.round()))

              precision    recall  f1-score   support

           0       0.79      0.67      0.73      4431
           1       0.72      0.83      0.77      4548

    accuracy                           0.75      8979
   macro avg       0.76      0.75      0.75      8979
weighted avg       0.76      0.75      0.75      8979
```



DecisionTreeClassifier:

```
[ ] #DecisionTreeClassifier
    from sklearn.tree import DecisionTreeClassifier
    DT_model = DecisionTreeClassifier()
    DT_model.fit(x_train,  y_train)
    DT_pred = DT_model.predict(x_test)
    acc_DT_model = metrics.accuracy_score(y_test,DT_pred)
    print("accuracy: ", acc_DT_model)

    accuracy:  0.7339347366076401
```

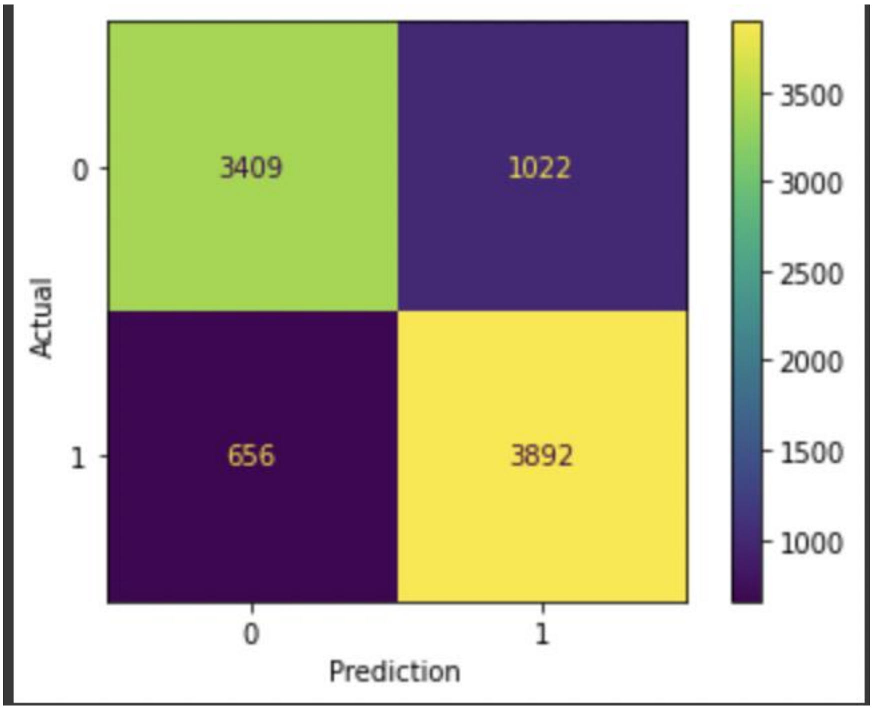
```
print(classification_report(y_test, DT_pred.round()))

              precision    recall  f1-score   support

           0       0.73      0.74      0.73      4431
           1       0.74      0.73      0.74      4548

    accuracy                           0.73      8979
   macro avg       0.73      0.73      0.73      8979
weighted avg       0.73      0.73      0.73      8979
```



RandomForest Classifier:

```
[ ] #RandomForestClassifier
    from sklearn.ensemble import RandomForestClassifier
    RF_model=RandomForestClassifier()
    RF_model.fit(x_train,  y_train)
    RF_pred = RF_model.predict(x_test)
    acc_RF_model = metrics.accuracy_score(y_test,RF_pred)
    print("accuracy: ", acc_RF_model)

    accuracy:  0.8131195010580243
```

```
print(classification_report(y_test, RF_pred.round()))

              precision    recall  f1-score   support

           0       0.84      0.77      0.80      4431
           1       0.79      0.86      0.82      4548

    accuracy                           0.81      8979
   macro avg       0.82      0.81      0.81      8979
weighted avg       0.82      0.81      0.81      8979
```



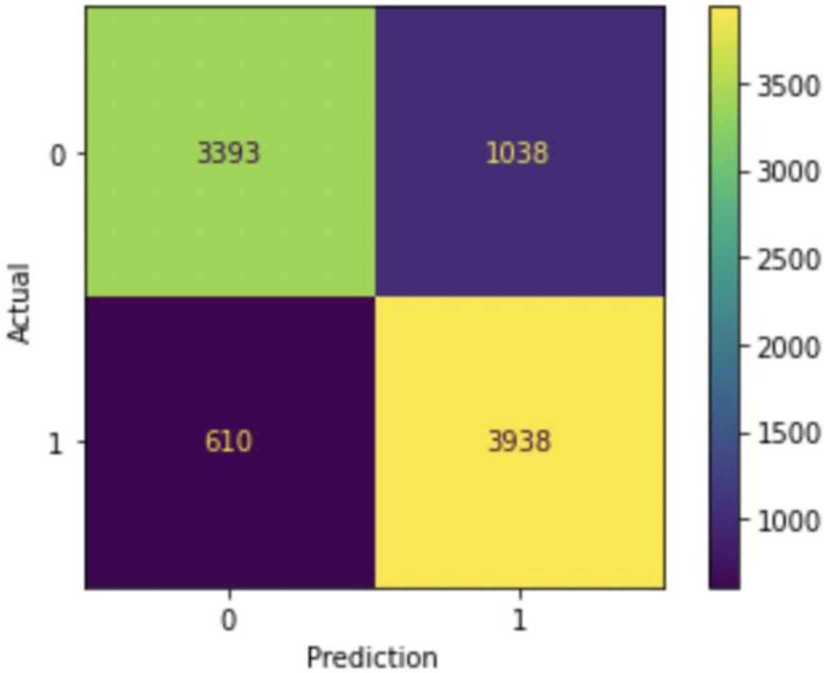
Xgboost:

```
# xgboost
import xgboost as xgb
xgb_model=xgb.XGBClassifier(learning_rate =0.01,n_estimators=200
xgb_model.fit(x_train, y_train)
xgb_pred = xgb_model.predict(x_test)
acc_xgb_model = metrics.accuracy_score(y_test, xgb_pred)
print("accuracy:", acc_xgb_model)
```

```
accuracy: 0.8164606303597283
```

```
[ ]
print(classification_report(y_test, xgb_pred))
```

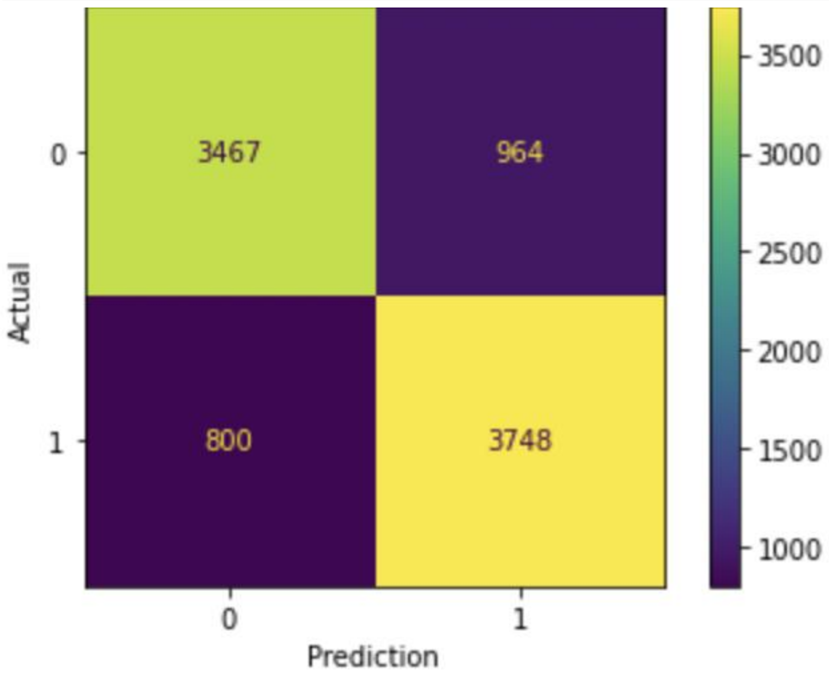
	precision	recall	f1-score	support
0	0.85	0.77	0.80	4431
1	0.79	0.87	0.83	4548
accuracy			0.82	8979
macro avg	0.82	0.82	0.82	8979
weighted avg	0.82	0.82	0.82	8979



Ensemble classifier1:

```
[ ] #Ensemble learning for classifier
    from sklearn.ensemble import VotingClassifier
    cl1= DecisionTreeClassifier()
    cl2 = RandomForestClassifier()
    cl3 = SVC()
    cl4 = KNeighborsClassifier()
    #cl5 = xgb.XGBClassifier()
    cl_voting =VotingClassifier(estimators=[('DT', cl1),('RF', cl2),('svm',cl3),('KNN', cl4)],voting='hard')
    cl_voting.fit(x_train, y_train)
    pred_all = cl_voting.predict(x_test)
    print(accuracy_score(y_test, pred_all.round()))

    0.8035415970598062
```

```
print(classification_report(y_test, pred_all.round()))

              precision    recall  f1-score   support

           0       0.81      0.78      0.80      4431
           1       0.80      0.82      0.81      4548

    accuracy                           0.80      8979
   macro avg       0.80      0.80      0.80      8979
weighted avg       0.80      0.80      0.80      8979
```

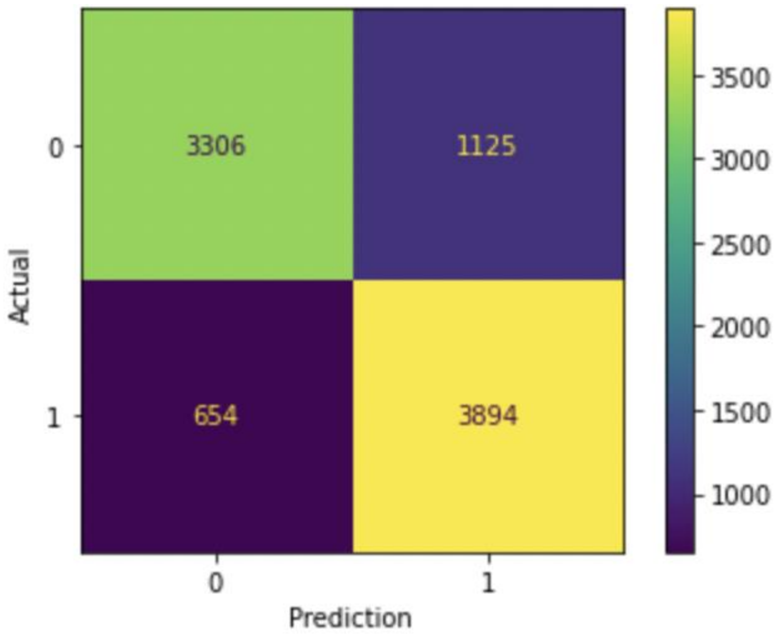


CatBoost:

```
accuracy: 0.8018710324089542
```

```
print(classification_report(y_test, Cat_pred.round()))
```

```
              precision    recall  f1-score   support

           0       0.83      0.75      0.79      4431
           1       0.78      0.86      0.81      4548

    accuracy                           0.80      8979
   macro avg       0.81      0.80      0.80      8979
weighted avg       0.80      0.80      0.80      8979
```



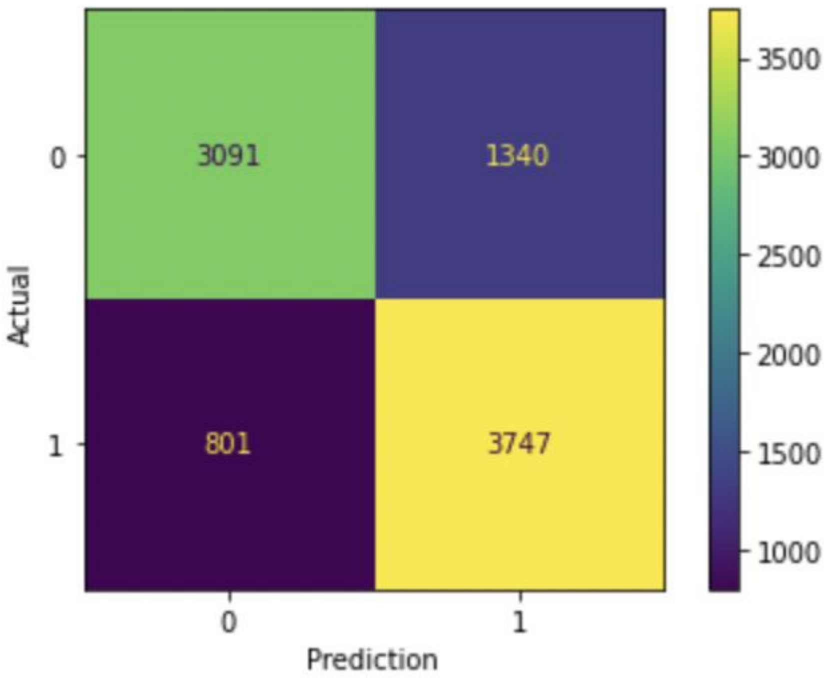
AdaBoost:

```
[ ] #adaboostclassifier
    from sklearn.ensemble import AdaBoostClassifier
    ada_model= AdaBoostClassifier(n_estimators=50,learning_rate=1)
    ada_model.fit(x_train, y_train)
    ada_pred = ada_model.predict(x_test)
    acc_ada_model = metrics.accuracy_score(y_test, ada_pred)
    print("accuracy:", acc_ada_model)
```

```
accuracy: 0.7615547388350595
```

```
print(classification_report(y_test, ada_pred.round()))
```

	precision	recall	f1-score	support
0	0.79	0.70	0.74	4431
1	0.74	0.82	0.78	4548
accuracy			0.76	8979
macro avg	0.77	0.76	0.76	8979
weighted avg	0.77	0.76	0.76	8979

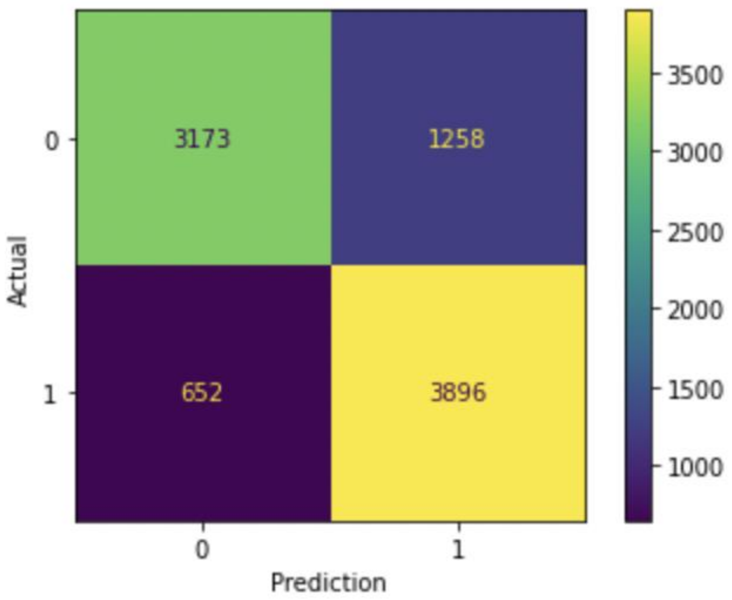


Ensemble learning for boostingclassifiers:

```
#Ensemble learning for Boostingclassifier
from sklearn.ensemble import VotingClassifier
cl5 = xgb.XGBClassifier()
cl6 = CatBoostClassifier()
cl7 = AdaBoostClassifier(n_estimators=50,learning_rate=1)
cl_boosting_voting =VotingClassifier(estimators=[('cat', cl6),('ada', cl7),('xgb',cl5)],voting='hard')
cl_boosting_voting.fit(x_train, y_train)
pred_boosting_all = cl_voting.predict(x_test)
print(accuracy_score(y_test, pred_boosting_all.round()))
```

```
print(classification_report(y_test, pred_boosting_all.round()))
```

```
              precision    recall  f1-score   support

           0       0.81      0.78      0.80      4431
           1       0.80      0.82      0.81      4548

    accuracy                           0.80      8979
   macro avg       0.80      0.80      0.80      8979
weighted avg       0.80      0.80      0.80      8979
```

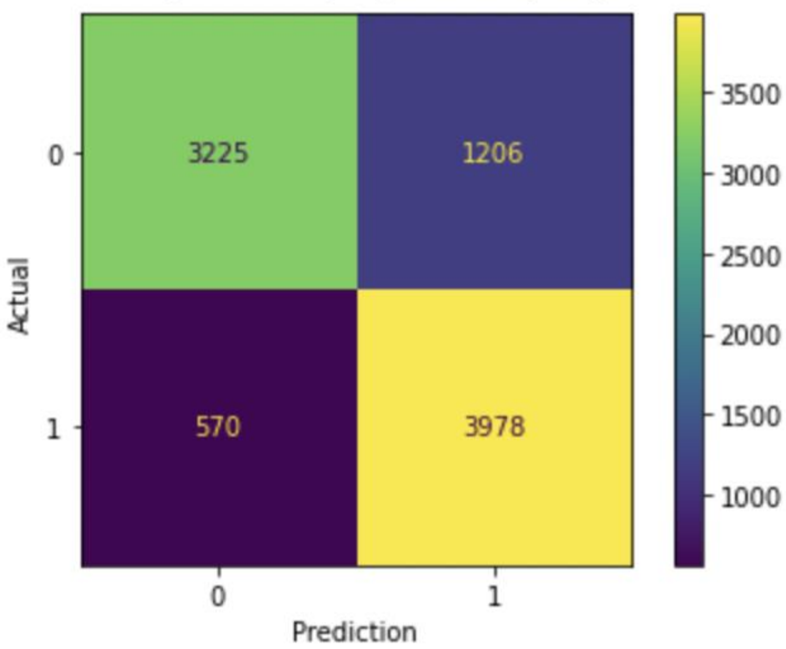


Ensemble learning for all models:

```
#Ensemble of all models
from sklearn.ensemble import VotingClassifier
cl1= DecisionTreeClassifier()
cl2 = RandomForestClassifier()
cl3 = SVC()
cl4 = KNeighborsClassifier()
cl5 = xgb.XGBClassifier()
cl6 = CatBoostClassifier()
cl7 = AdaBoostClassifier(n_estimators=50,learning_rate=1)
cl_all_voting =VotingClassifier(estimators=[('DT', cl1),('RF', cl2),('svm',cl3),('KNN', cl4),('cat', cl6),('ada', cl7),('xgb',cl5)],voting='hard')
cl_all_voting.fit(x_train, y_train)
pred_complete = cl_all_voting.predict(x_test)
print(accuracy_score(y_test, pred_complete.round()))
```

```
print(classification_report(y_test, pred_complete.round()))
```

```
              precision    recall  f1-score   support

           0       0.85      0.73      0.78      4431
           1       0.77      0.87      0.82      4548

    accuracy                           0.80      8979
   macro avg       0.81      0.80      0.80      8979
weighted avg       0.81      0.80      0.80      8979
```



Comparative Study:

Models	Accuracy
Naïve bayes	71.31
SVM	77.61
KNN	75.11
Decision Tree	74.17
Random Forest	81.57
Ensemble 1	80.32
XGBoost	81.73
AdaBoost	76.05
CatBoost	80.28
Ensemble 2	80.47
Ensemble 3	80.09

Conclusion:

From the above results and discussion we can say that XGboosting and Catboosting gives more accuracy and there ensemble model also gives good accuracy result.

IX.References:

1. Understanding Musical Predictions With An Embodied Interface For Musical Machine Learning Charles Patrick Martin, Kyrre Glette, Tønnes Frostad Nygaard And Jim Torresen.

2. Deep Learning In Music Recommendation Systemsmarkus Schedl.

3.Applications Of Machine Learning To Music Research: Empirical Investigations Into The Phenomenon Of Musical Expression By Gerhard Widmer.

4. Deep Learning For Music By Allen Huang, Raymond Wu.

5.Music Rhythm Tree Based Partitioning Approach To Decision Tree Classifier By Shankruguggari, Vijayakumar Kadappa, Umadeviv, Ajithabrahamb.

6.Analyzing And Predicting Songs Using Machine Learning Techniques By Mr.J.Tulasi Ram,M.Tech 1 N. Bhavyasai2 ,K. Ankitha Lavanya 3 , K. Vaishnavi Tulasi 4 , K. Kowshik Varma 5

7.Examensarbete Inom Teknik, Grundnivå, 15 Hp Stockholm, Sverige 2018 Predicting Hit Songs With Machine Learning Minna Reiman Philippa Örnell.

8. Experimenting With Music Taste Prediction By User Profilingmarco Grimaldi, Pádraig Cunningham.

9. A Music Recommendation Algorithm Based On Clustering And Latent Factor Model Yingjie Jin And Chunyan Han*.

10. A Model For Predicting Music Popularity On Streaming Platforms Um Modelo Para Previsao Da Popularidade De M ´Usicas Em Plataformas De ˜ Streaming Carlos V. S. Araujo1 *, Marco A. P. Cristo1 , Rafael Giusti1.