**RESONANCE - AN AI BASED MUSIC RECOMMENDATION SYSTEM**

Yashvi Kommidi  *Hyderabad, India*  
[yashvikommidi31@gmail.com](mailto:yashvikommidi31@gmail.com)

Thudi Bhanu Prasad *Hyderabad, India*  
thudibhanuprasad4@gmail.com

Masula Keerthi *Hyderabad, India*  
keerthimasula2712@gmail.com

***Abstract*—** *The music recommendation engine, for example, is a specific type of recommendation system that anticipates consumer preferences and recommends songs to the subscriber according to a variety of factors. The amount of music that are accessible exceeds a person's lifetime listening capacity and keeps growing every day. A person may find it challenging to choose from the millions of songs available, and there is a good chance that they will overlook some of their potential favourites. Personalization algorithms and endless streaming services have replaced Radio DJs as the gatekeepers of music since their heyday. Nowadays, the general public listens to music from all genres because it is so easy to access it globally. However, there are a number of areas where current algorithms fail. Without enough historical data, how could an algorithm determine as to if listeners will enjoy a brand-new song or performer? And how does it choose the music that it recommends to new users? Here, we'll develop a recommendation system that, a month after a user's first audible listening event, can predict that user's subsequent musical listening behaviour. Our strategy will largely depend on systematic and thorough feature engineering and a variety of boosting classification techniques, which are both immediately practical in the real world.*

***Keywords— Recommendation System, Machine Learning, XG Boost***

# I.INTRODUCTION

Streaming services for digital content are increasingly popular in the modern era. This is particularly apparent in the world of streaming music. Millions of active users are said to be using Spotify, Pandora, and Apple Music, and those numbers are continually rising. Because it's so simple to choose which music to listen to, most people prefer streaming services over radio stations because it's much more than having a music library on CDs. Users, however, have a very difficult time locating pertinent material in such a vast array of items. As a result, service providers work to easily gain and specialised ways to find music. According to the MarsBands.com, there really are 97 million music in existence. We would reach 200 million songs if we featured well-known music or  incredibly old Celtic songs without names, as the web page is unlikely to contain Happy Holidays or an unnamed song from 1400 AC. Only artists whose identities have been expressly enumerated on music charts are included here. For the sake of this discussion, let's assume that there are one million active songwriters at the moment. If we use the same proportion as above, we can calculate the number of roughly 15.3 million composers throughout history.

4 million tracks on Spotify have not been played, to give you an idea. The number of songs available must be in the billions, and Spotify is not by any implies the end of the world of music. Consider all of the CDs as well as records created so over past century but not preserved digitally? How about African song traditions that have been handed down through the generations? There are trillions upon trillions of songs in existence, making it impossible to count individuals all. It's possible that there are an infinitely greater amount of unwritten songs.

Recommender systems, which only suggest content that consumers are most likely to find useful, help to solve this problem to a large extent.

A tiny proportion of recommender system called music recommendation systems forecast user preferences and make strong song recommendations based on a variety of variables. They are sharp on both edges. Both the provider and the user can profit. By restricting the options available, they make things easier for the user and keep them engaged by asking for recommendations for good music. They provide options for exploring and discovering music that the user may not be conscious of. Since it recommends music, there is always entertainment.These address the issue of having to choose from a variety of options, as there is occasionally a risk of skipping songs that might have been favourite tracks.

## 1.1 Existing System

All recommendation models in the current system are created solely based on past usage, and users see a showcase of the tunes in their music library depending on the past history of songs they have listened to.

Major primary drawback of the existing arrangement is that it does not analyse songs to determine how many times users have listened to them repeatedly. It's also unable to appropriately highly suggest songs to new users. A composer might find himself in this situation where he is unsure of whether his song will be a success or a failure.

## 1.2 Proposed System

Both new and experienced users can receive accurate song recommendations from our envisioned system. It has the ability to foretell will a user relisten to same song repeatedly during a specified period of time.

This aids a composer in analysing his or her song so that they can write new songs based on the conclusions they have reached.

# II. Literature Survey

A perfect songs recommender system has capability to automatically suggest personalised songs based on their listening experiences to listeners. Millions of users have currently signed up for various music-discovery websites like Spotify, Apple Genius, All Songs, Pandora, Audio Bhai Mog, and last.fm. [5] In this section, we present an overview of the most significant methods, including collaborative filtering, context-based retrieval of information, emotion-based modelling, and hybrid models.

Users in groups with similar interests are given song suggestions using the collaborative filtering method[11]. The benefit of using collaborative filtering is that song recommendations can be customised. The 3 kinds of recommendation collaborative filtering techniques that are implicated are memorybased histogram equalization, modelbased collaborative filtering, and hybridbased collaborative methods.

The recommendation systems currently in use have had great success with their collaborative filtering algorithms[10][6][7]. The efficient algorithm, which utilised significant factor models, outperformed Netflix's current algorithm by 11.09% when it launched a competition to determine the best cooperative filtering algorithm. Amazon uses collaborative filtering[3] among users and products, which significantly contributes to the success of the business. The more recent neural collaborative filtering algorithm makes use of neural networks.

The content-based filtering method forecasts the song by looking at the song track. It suggests songs that are similar to songs the user has already started listening to instead of songs which the user has rated "like" on the basis of information retrieval and filtration. Extracting and comparing acoustic features to find tracks that are perceptually similar has been the major aim of numerous studies. The ones that stand today out most are timbre and rhythm. Based on the features that were extracted, the ranges between songs are calculated. Average Feature, Anticipation with Monte Carlo Sample selection, and K-Means Clustering with Earth-Distance Mover's Three typical methods used to determine similarity are vectors with Euclidean distance.

For content-based algorithms, many researchers[4][7][18] have proposed a variety of methods using machine learning techniques, such as various regression analysis, support vector machines, and decision trees. Using the information we learned in class, we can fully implement these algorithms[17].

## The global digitization of music has made it easier to access different musical genres[12]. It is inconceivable to find the opportunity to listen to and evaluate tunes in order to accumulate a personal entire music collection because of the growing amount of work. Making a search engine or music recommendation engine based on different moods could be one solution.Combining a variety of aesthetic and semantic features taken from textual lyrics to produce a mood classification system.

# III. Dataset Features

## Over 700,000 user-user interactions, 113,750 songs, and 3,076 users are all represented in our dataset. It has no audio features and only text data. The binary values for this dataset's class labels—whether the user pays attention to music again—are roughly balanced: In 50.9% of cases, labels are positive, while in 49% of cases, they are negative. The raw dataset includes 11 categorical attributes and 6 numerical features, with categorical features making up the majority of the features. User id and song id have such a high cardinality, whereas lyricist is a highly rare feature. After performing feature extraction, we split the data into three sets: train, validation, and test, with respective percentages of 70%, 15%, and 15%.

# 

# IV. Software And Hardware Requirements

The two main categories of requirement analysis:

### 1. Software Requirements

The following list includes the software requirements for the project.:

Python programming language

Operating System:Windows 7 and above

### 2. Hardware Requirements

The most crucial hardware requirements for the project are listed below.

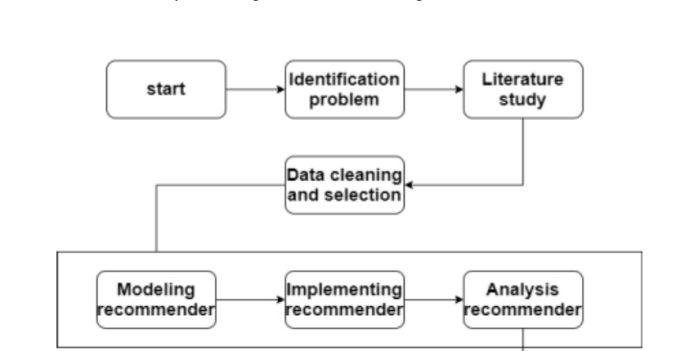
Storage: 500 GB

RAM minimum 4 GB

# V. Methodology

We discuss the design of the system in great detail in this section.

The outcome of the training process is a model of machine learning which is referred to as the arithmetical depiction of the real practical examples. The algorithms search for patterns in the training data, which is utilized to approximate the candidate solution and is in control of tracing the inputs towards the outputs from the dataset obtained. [19] Depending on the task at hand, these machine learning techniques are grouped as Classification techniques, Regression methods, Clustering, Computational complexity Reductions, Principal Component Analysis etc.

Automation is no exception and necessitates a sufficient flow of organised, diverse data to be effective. Businesses today have access to an enormous amount of data about their consumers, typically millions of pieces of information. Because of the abundance of information it contains, the aforementioned data, which really is big in terms of the number of large datasets and the number of fields, is did refer to as big data.****

**Figure 3.1: System Diagram**

## There are several stages to our research, of which the first is the beginning. It includes identifying the issue and reviewing the literature. a model-recommendation system, developing, utilising, and evaluating the software Analysis and Recommendation of the music-recommender system.

## Data Preprocessing

### The flow of replacing, editing, deleting the inaccurate or coarse data is known as data cleaning and involves identifying any incomplete, misleading, or removing ) corrupt or inaccurate. Finding and fixing (or removing) inaccurate or corrupt records from a national record, table, or database is the process of "data cleaning."

### Data selection: The process of selecting the best source of data, type, and tools for data collection is known as data selection. Data selection takes place before actual data collection. Choosing the right information for a study project can have an impact on data integrity.

### Feature Extraction

We clearly and unambiguously infuse the cell - associated by cluster analysis the set of data by user id and normalising the row indices because the user behavior information is recorded in chronological and the timestamp may be an important factor in determining user preference. We use Label Encoder to convert text values for characteristics like composer identities, songwriters, and user cities into integer labels because they are categorical values even though they are text values. Additionally, all of the numerical features were standardised..

For gradient enhancing and decision tree models, date columns such as user account registration date and email address expiry date are converted to date count. This foundation date was arbitrarily selected because the earliest account creation date is in 2004, but it has no effect on how well tree-based models work.

# VI. Model Building

### The feature extraction method produces extremely high dimensional data, which is reduced using classification. In order for the algorithms for machine learning to reason about artefacts belonging to different groups, features must take a variety of values.

The following techniques can be used to evaluate the system:

## 6.1 Decision Tree

The prediction and classification tools known as decision trees are efficient and well-liked. Decision trees portray rules that humans can comprehend and can be employed in scientific knowledge like databases. The local region is identified in fewer steps through a series of iterative splits using a hierarchical model for reinforcement methods known as a decision tree. A decision tree is made up of internal decision points, internal decision endpoints, and terminal leaves. Each decision node m implements the test function fm(x), and the branches are labelled using discrete results. Every node tests an input, and based on the results, it chooses one of the branches.A decision tree is also a quasi model because we do not even make the assumption any parameterized shape for the category densities and the binary tree is really not fixed beforehand; instead, it changes throughout learning depending on how complex the problem is inherently in the data.

## 6.2 Random Forest

Random forests is a classification ensemble technique. In the Random Forest prototype, multiple trees are planted as opposed to just one tree inside the Decision Tree model. However, the issue of why utilise multiple decision trees when a mature plant can actually achieve the same job at hand has been raised. Overfitting is a major problem with decision trees because it produces a very subpar predictive model. A random forest with more trees produces a much better prediction models by removing overfitting. We say that a tree "votes" for a class when it offers a categorization to characterise a new item based on attributes.

*6.3 AdaBoost*

The most popular boosting algorithm for classiﬁcation is called Ada Boost, and it was developed by Freund and Schapire. For binary classification problems, it requires a training set "S" of "m" examples (S=(x1,y1),...,(xm,ym)), where each instance (example) xi is a vector of feature values belonging to a domain or instance SpaceX so each label yi is the class label affiliated with xi which belongs to a finite label space Y=1, +1.

*6.4 XG Boost*

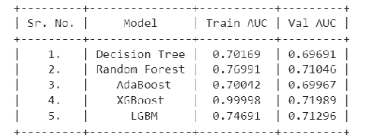
XG Boost is a well-liked and practical obtainable interconnection of the contour-boosted trees algorithm (eXtreme Gradient Boosting). In order to accurately estimate a target variable, the supervised learning algorithm gradient boosting combines a number of estimates from various weaker and simpler models. Because of the algorithm's rigorous handling of a wide range of information types, relationship issues, variances, and the diversity of model parameters that are available to fine-tune, the XG Boost system computes well in computational competitions. Analysis of variance, characterization (binary and multiclass), and ranking issues can all be solved using XG Boost.

*6.5 LGBM*

In fact, Light GBM is a decision tree-based back- propagation learning framework that enhances model performance while utilising less memory.

It uses two novel techniques: longitudinal stripe one side sampling and exclusive feature cobbling, to address the shortcomings of the histogram-based algorithm, which is mainly used in all gradient descent algorithms (Gradient Boosting Decision Tree) frameworks (EFB). The GOSS and EFB methodologies that are used in the Light GBM Algorithm to create its characteristics are described below. Together, they give the model an edge over other GBDT architectures and allow it to operate efficiently.

# Vii. Results and Discussion



**Figure 3.2: Models Comparisio**

After training our dataset with all the mentioned models, XG Boost turned out to be the most efficient model with 0.99 Train Accuracy and 0.71 Val Accuracy

# Viii. Conclusion

Our findings from the experiment are summarised below. To enhance the accuracy of music recommendations, music recommender systems should first take music genre information into account. Based on the features of the songs, the music recommender can make song recommendations. The song recommender can anticipate whether the consumer will listen to the melody again soon. Even though different music recommendation system operate in various ways, the complexity of machine learning systems such as the Music Recommendation System prevents them from having a standardised structure. Our analyses allow us to recommend additional music features for future research in order to increase the recommender system's accuracy, including the use of tempo gramme to record the local tempo at a specific time.

Given user listening background and musical information, extreme gradient boosting generally yields the best results for predicting repeated music listening behaviour.

# Ix. Future Scope

This can be done in order to increase prediction accuracy even further. Matrix factorization can be used to translate categorical features to numerical features. It is possible to use log transformation to enhance heavy-tail features. It is possible to implement outlier detection to actively remove less useful data. Various modelling methodologies can be tested. Compared to the current ensemble boosting model, stacking might yield more intriguing results. To improve the results of stacking models, more models of different types, even if they are individually slightly less accurate, can be added. Furthermore, there is still room for advancement with the random forest model.

##### x. References

**[1]** D. Arditi, Digital Subscriptions: The Unending Consumption of Music in the Digital Era, Popular Music and Society (2017).

**[2]** Choi K, Fazekas G, Sandler M. Automatic Tagging using Deep Convolutional Neural Network. arXiv eprints arXiv:1606.00298. 2016.

**[3]**Wang X, Wang Y. Improving Content-based and Hybrid Music Recommendation using Deep Learning. In Proceedings of the 22nd ACM international conference on Multimedia; 2014; Orlando. p. 627-636.

**[4]**Everyone listens to music, but how we listen is changing. [online] Available at: http://www.nielsen.com/us/en/insights/news/2015/everyone-listens-to-music-but-how-we-listen-is-changing.html [Accessed 10 Oct. 2017].

**[5]**Labrosa.ee.columbia.edu. (2017). Million Song Dataset | scaling MIRresearch. [online] Available at: https://labrosa.ee.columbia.edu/millionsong/ [Accessed10Oct. 2017].

**[6]**.Linden, G., Smith, B. and York, J. (2003). Amazon.com Recommendations Item-to Item Collaborative Filtering. [ebook] Available at:https://www.cs.umd.edu/~samir/498/Amazon-Recommendations.pdf [Accessed10Oct. 2017].

**[7]** Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-SengChua. 2017. Neural Collaborative Filtering. In Proceedings of the 26th International Conference on World Wide Web (WWW '17). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 173-182. DOI: https://doi.org/10.1145/3038912.3052569.

**[8]** Gershman, A. and Meisels, A. (2015). A Decision Tree Based Recommender System.[ebook] IEEE Xplore, pp.170-179. Available at: https://subs.emis.de/LNI/Proceedings/Proceedings165/170.pdf [Accessed 9 Oct. 2017].

**[9]**Min SH., Han I. (2005) Recommender Systems Using Support Vector Machines. In: Lowe D., Gaedke M. (eds) Web Engineering. ICWE 2005. Lecture Notes in Computer Science, vol 3579. Springer, Berlin, Heidelberg.

**[10]**R-bloggers. (2017). Hybrid content-based and collaborative filtering recommendations with {ordinal} logistic regression (2): Recommendation as discrete choice. [online] Available at: https://www.r-bloggers.com/hybrid-content-based-and-collaborative-filtering-recommendations-with-ordinal-logistic-regression-2- recommendation-as-discrete-choice/ [Accessed 7 Oct. 2017].

**[11]**Choi K, Fazekas G, Sandler M. Convolutional Recurrent Neural Networks For Music Classification. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

**[12]** Schedl M, Zamani H, Chen CW, Deldjoo Y, Elahi M. Current challenges and visions in music recommender systems research. Int J Multi Inform Ret. (2018) 7:95–116. doi: 10.1007/s13735-018-0154-2

**[13]** Schedl M, Gómez E, Urbano J. Music information retrieval: recent developments and applications. Found Trends Inform Ret. (2014) 8:127–261. doi: 10.1561/1500000042

**[14]** Downie JS. Music information retrieval. Ann Rev Inform Sci Techn. (2003) 37:295–340. doi: 10.1002/aris.1440370108

**[15]** Dorfer M, Henkel F, Widmer G. Learning to listen, read, and follow: score following as a reinforcement learning game. In: Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR 2018), Paris (2018). p. 784–91.

**[16]** Chou PW, Lin FN, Chang KN, Chen HY. A simple score following system for music ensembles using chroma and dynamic time warping. In: Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval (ICMR 2018), Yokohama: ACM (2018). p. 529–32. doi: 10.1145/3206025.3206090

**[17]** Goto M, Dannenberg RB. Music interfaces based on automatic music signal analysis: new ways to create and listen to music. IEEE Signal Proc Mag. (2019) 36:74–81. doi: 10.1109/MSP.2018.2874360

**[18]** Schedl M. Intelligent user interfaces for social music discovery and exploration of large-scale music repositories. In: Proceedings of the 22nd ACM International Conference on Intelligent User Interfaces (IUI 2017): Workshop on Theory-Informed User Modeling for Tailoring and Personalizing Interfaces (HUMANIZE 2017). Limassol: ACM (2017). p. 7–11. doi: 10.1145/3039677.3039678

**[19]** Oramas S, Nieto O, Barbieri F, Serra X. Multi-label music genre classification from audio, text and images using deep features. In: Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR), Suzhou (2017). p. 23–30.

**[20]** Mayer R, Rauber A. Music genre classification by ensembles of audio and lyrics features. In: Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011). Miami, FL (2011). p. 675–80.

**[21]** Sturm BL. Classification accuracy is not enough: on the evaluation of music genre recognition systems. J Intell Inform Syst. (2013) 41:371–406. doi: 10.1007/s10844-013-0250-y 2016.