Effective Music Recommendation System Employing Machine Learning/Deep Learning Techniques

Diego Landi
Electrical and Computer Engineering
Florida International University
Miami, United States
dland034@fiu.edu

Carlos Otero
Electrical and Computer Engineering
Florida International University
Miami, United States
coter025@fiu

***Abstract*—In recent years typical music consumption has significantly changed. Personal music collections have grown, aided by technological improvements in networks, storage, portability of devices, and personalized internet services such as music recommendation systems (MRSs). A recommendation system is a program that utilizes techniques to suggest to its users' items that would likely match their preferences. This project focuses on improving MRSs by testing distinct machine-learning and deep-learning algorithms, comparing accuracy metrics for each model, and narrowing down the optimal model that more accurately trends the individual's music preference and provides real-time predictions. The team applied this approach to data obtained from Kaggle, attaining precision scores as high as 82**

***Index Terms*—Music recommendation, machine learning, deep learning, optimal model**

I. Introduction

Music is regarded as an essential aspect of people's lives, as showcased in previous research that indicates that participants listen to music more often than any other entertainment activities such as watching television, reading books, and watching movies [1]. The widespread usage of the Internet has brought about relevant modifications in the music industry regarding online music listening, control of music copyright, classification of music genres, and music recommendations. Nowadays, with the development of music broadcast platforms, individuals can listen to music anytime and anywhere within a platform containing multiple songs, such as Spotify and Apple Music. Along with the rapid advancement of music platforms, the expansion of digital music formats, managing and searching for songs enabled the improvement of music information retrieval (MIR) techniques. Though recent research presented the success of MIR techniques [2], the development of music recommendation systems is still in progress [3]. Automatically recommending music utilizing audio properties has attracted significant attention from researchers [4], [5], but even the best systems still do not match expectations.

There are two typical approaches to developing a recommendation system: content-based and collaborative. Content-based techniques analyze the contents of objects previously liked by the user and recommend the ones that fit the individual liking accordingly. Suppose a user likes A, and B is similar to A. In that case, a user might like B. Alternatively, collaborative methods base their recommendations on what user groups with similar preferences have liked. For instance, if user A is similar to user B, then user A might like what user B likes. Making the collaborative technique less reliable in some situations since it does not consider the individual's particular preference [6].

To analyze individual preferences, the team utilized a dataset consisting of user-liked/disliked songs. Each song has 14 acoustic attributes that will be utilized to generate a mathematical model that trends the individual's music preference. For the model generation, the team utilized different machine learning algorithms such as Decision Tree Classifier, KNN, Support Vector Machine, Random Forest Classifier, and distinct deep learning architectures to obtain the algorithm that optimally defines the user's preference. Furthermore, the team utilized an available library, GridSearchCV, to iterate different parameters of the machine-learning algorithms to obtain the best model for each tested technique. By comparing distinct algorithms results, the team could more accurately assert if the music inputted in the model matches the user's preference.

II. Literature Review

Current research has explored the utilization of machine learning and deep learning algorithms to improve current music recommendation systems and provide more reliable predictions regarding users' music preferences. The work published by Dong-Moon Kim et al. elaborates on their approach to developing an accurate music recommendation system by a dynamic K-Means Clustering algorithm. As mentioned in the article, "To analyze a user's preference, we save the list of music the user downloaded or listened to and infer the user's intention from the list. We propose a dynamic K-means clustering algorithm to analyze users' preferences. The number of clusters, k, should be given in advance for k-means to work correctly. So, if k is given correctly, the data in the list maybe properly grouped, and the grouped music can be used as the user's preference [6]". By utilizing this method and finding the optimal number of clusters for accurate real-time predictions, the researchers obtained an accuracy score of 80%.

In this context, Ahmet Elbir et al. proposed in their article utilizing machine learning algorithms such as KNN, Random Forest, and Support Vector Machine, combined with acoustic features present in songs to not only recommend but also to classify the song genre. The researchers elaborated a unique technique to obtain the best features contributing to a more robust model generation. As mentioned in the article, "The parts to be extracted from the music have been determined as zero crossing rate, spectral centroid, spectral contrast, spectral bandwidth, spectral roll off and Mel-frequency Cepstral Coefficients-MFCC. Furthermore, Convolutional Neural Network-CNN, one of the most useful deep learning methods, has been used for music genre classification and recommendation [3]". By executing this unique feature engineering method, the researchers obtained a highly accurate model that classifies songs by genre and recommends their likelihood to be suggested to the specific user.

Along these lines, Ferdos Fessahaye et al. elaborated on their novel music recommendation system using deep learning techniques in their publication. As mentioned in the article, "We created a novel algorithm for recommending music to users in the form of k recommendations. Our system, Tunes Recommendation System (T-RECSYS), scores each song in a database according to user preference by utilizing a learned hybridization of content-based and collaborative filtering, returning the top-k scoring songs. The algorithm incorporates user input using historical data and preferences, extracting multiple key metadata variables such as genre, mood, and tempo [7]". By combining both recommendation techniques (content-based and collaboration) into a single deep-learning architecture, the researchers obtained accuracy scores when recommending songs to users up to 88%.

Finally, Mohammad Tabrez Quasim et al. proposed an emotion-based music recommendation system that incorporates machine learning techniques with IoT frameworks in their article. Their approach to developing this system is described as follows, "Firstly, researchers selected six categories of songs that comprise the simple database: joyful, silent, emotional, depressed, inspired, and enthusiastic. In this document, researchers retrieve the Mel Frequency Coefficient function's magnitude, cluster it and get the soundtrack's worth. Then researchers place information in the repository, including the artist, the song's description, and their function score. If the person enters the suggested sound, the same functionality attribute is derived, comparing the song's length in the collection. If the subjective area is close, all pieces are content-like: the source song's suggestion [8]". By incorporating this approach, researchers could generate models with a prediction accuracy of 96.12%.

It is observable from recent developments in this subject that utilizing machine learning and deep learning algorithms to recommend music to users is feasible and returns satisfying results when compared to other traditional statistical approaches, but there is still room for improvement.

III. Machine Learning Algorithms

In this project the usage of different algorithms was key to determining which approach would provide the highest accuracy when recommending a song. In this experiment a total of six different algorithms were implemented. The classifiers that were used are the following: Multi-layer Perceptron (MLP) Classifier, Gaussian Naive-Bayes Classifier, KNeighbors-Classifier, Decision Tree Classifier, Support Vector Machines (SVM) Classifier, and Random Forest Classifier. In addition, each one of these classifiers has different advantages and disadvantages that can come into play if any of the classifiers have similar performances.

A. MLP Classifier

The MLP Classifier is a supervised learning algorithm that learns a function by training on a dataset using backpropagation [9]. The MLP Classifier is a neural network composed of an input layer that feeds to a set of hidden layers followed by an output layer. The main advantages of this algorithm is its capabilities to learn non-linear models and learn models in real time [9]. This means that the MLP Classifier can develop decision boundaries for problems that are not linearly separable. The disadvantages of this algorithm are its sensitivity to feature scaling, the requiring of a number of hyperparameters to be tuned and having a non-convex loss function [9]. This means that sometimes values for the hyperparameters may not actually provide satisfactory results. An example of such an issue can be seen with the different learning rates. If the learning rate is large the descent of the loss function is very steep at first but then it reaches a point where the loss functions oscillates back and forth between two values. In contrast if the learning rate is very small the loss function has a gradual decent but it eventually can lead to overfitting.

B. Gaussian Naive-Bayes Classifier

The Gaussian Naive-Bayes Classifier is also a supervised learning algorithm that utilizes Bayes theorem under the "naive" assumption of conditional independence between every pair of features and the assumption that such pair of features distribution is Gaussian [10]. The Naive-Bayes Classifier umbrella has many different algorithms that make different assumptions pertaining to the distribution of the dataset. This means that a correlation between the performance of the classifier and assumptions can be established. The advantages of such a classifier is the fact that its extremely fast at its task [10]. Hence, this algorithm accuracy can serve as a great litmus test to identify which classifiers are worth investigating further. However, since this classifier is known to be bad estimators (i.e their probability outputs are not to be taken seriously) then the overall performance of the classifier must be considerably higher than other classifiers in order to make up for this deficiency [10].

C. K-Neighbors Classifier

The K-Neighbors Classifier is an instance based learning that utilizes a simple voting system to assign a data class based on how many representatives of the class is within the nearest neighbor [11]. Essentially, this classifier devices its decision boundaries by identifying the distances between data points. The benefits of such a learning strategy include suppressing the impacts of noise in a dataset by selecting a bigger k value [11]. This is particularly useful since all datasets contain noise which means that by suppressing its impact on the dataset, the accuracy and performance of the classifier is improved. However, the downside of selecting a bigger k value is that classification boundaries are less distinct [11]. A less distinct decision boundary can be interpreted as the classifier not being able to generalized which means that the model is essentially in some of overfitting.

D. Decision Tree Classifier

The Decision Tree Classifier uses a non-parametric supervised learning method to make the decision rules on a given dataset [12]. As opposed to the classifier being defined by function that demonstrates the decision boundary, Decision Tree can implement the features as rules in order to make decision and classify the dataset. The advantages of such an algorithm is that they require little data preparation, are simple to understand and the cost of using them is logarithmic in the number of data points used to train the tree [12]. In this regard they can be very enticing to implement into projects because they eliminate lots of the issues that are prevalent in most machine learning algorithms. However, their downside is that they can be unstable due to small variations and the fact that there predictions are piece-wise constant approximations [12]. Therefore, the ability to predict future occurrences from previous data is severely compromised due to the classifier not being able to make predictions that are smooth and continous approximations.

E. SVM Classifier

The SVM Classifier is supervised learning method whose decision function depends on some subset of the training data known as the support vectors [13]. Since these support vectors depend on the subset of the training data, it is possible to obtain lots information regarding the dataset. The advantages of such an algorithm is that is effective in high dimensional spaces, it is memory efficient and incredibly versatile due to the different kernel functions [13]. The ability to be memory efficient while tacking high dimensional speaces is a crucial when it comes to machine learning algorithms since many problems can have a large amount of features. However, it lacks the ability to directly provide probability estimates and can is sensitive to the kernel functions [13]. This means that when designing this type of classifier it is important to iterate through all of the kernels to obtain the best one.

F. Random Forest Classifier

The Random Forest Classifier is an averaging algorithm based on randomized decision tree [14]. The averaging algorithm on decision trees basically means that this classifier gives access to some of the benefits of decision trees while eliminating some downsides. One benefits of this algorithm is the reduction in variance due to the combination of trees, however this can result in a slight increase in bias [14]. This means that the instability in decision tree due to small variations can be mitigated by the usage of this algorithm. However, this at the expense of introducing bias which can affect the decision rules that are implemented in Decision Tree Classifier.

IV. Results and Discussion

The purpose of this project is to compare the results from the different classifiers and determine the best performing. However, since each classifier has different hyper-paremeters it would not be prevalent to compare only the results of one set of hyper-parameters for each individual classifier. Instead GridSearchCV will be implemeneted in order to iterate through different combinations of hyper-parameters.

```
parameters_randomf = {'criterion':('gini','entropy','log_loss'),
                      'n_estimators':[x for x in range(1,50)]}
random_forest = RandomForestClassifier()
grid_randomf = GridSearchCV(random_forest,parameters_randomf)
optimal_randomf = grid_randomf.fit(Xtrain_N,y_train)
```

Fig. 1. Code for Random Forest Algorithm

In Fig. 1 it can be observed that the program is assigned a set of selected hyper-parameters i.e the 'criterion' & 'n_estimators' with a list of possible values for these hyper-parameters. This information is then passed alongside the Random Forest Classifier to the GridSearchCV function to determine the best combination. After obtaining the best combination of parameters for each given classifier, the network is trained and scored in order to determine the accuracy of the classifier.

TABLE I
ACCURACY OF EACH CLASSIFIER

Classifier	Accuracy
MLP	73 %
Gaussian Naive-Bayes	66 %
KNeighbors	71 %
Decision Tree	72 %
SVM	75 %
Random Forest	82 %

In Table I the highest accuracy achieved was with the Random Forest algorithm, while the lowest accuracy was obtained by using the Gaussian Naive-Bayes Classifier. In this aspect there are some key information that can be hypothesized. If the Gaussian Naive-Bayes Classifier has such a low accuracy it must be that the assumption of conditional independence and/or Gaussian distribution are not valid. In order to determine the validity other experiments can be conducted using different types of Naive-Bayes algorithms. Once it was determined that the Random Forest Classifier performed the best it became necessary to develop a classification report and confusion matrix. These tools will allow for a further examination of how the Random Forest is Classifying the data.

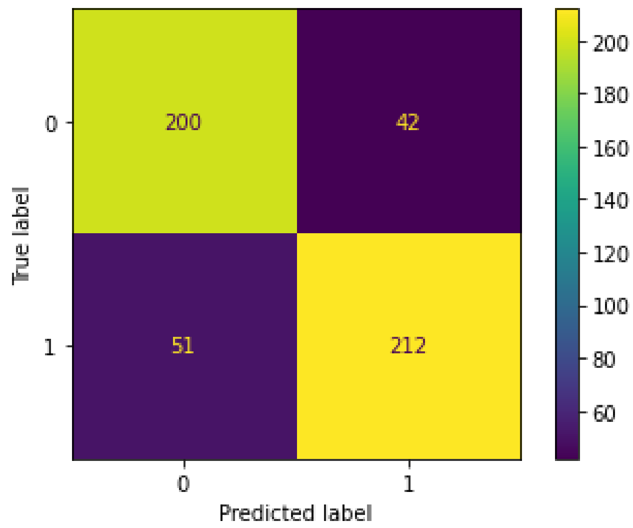

Fig. 2. Random Forest Confusion Matrix

In Fig 2 the information presented allows for an understand key features into how the random classifier is performing. For example, when it comes to music recommendation its important that the precision of identifying the music the individual will like be higher than its complement. From the Fig 2 its calculated that precision of the classifier at predicting that user will like the song is an 83%, while its complement its an 80%. Therefore, once the information has been understood, its time to use the Random Forest Classifier to organize a master database comprising of 170653 songs from the highest to lowest probability of the certain individual liking the song.

	artists	id	name	Probabiltiy
148465	['Patrice Rushen']	37VzvL5Zu21ogLgKm0J8lY	Get Off (You Fascinate Me)	1.0
100245	['Meco']	1CarOKwggz4XkxFfr09sRp	Medley: Star Wars - 12" Version	1.0
101009	['Bohannon']	4fW5B7Dp2hJoyMKbg4tkxg	Take the Country to New York City	1.0
100877	['Bernard Wright']	0tUYG0VaHjQDbdNQFawtDY	Spinnin'	1.0
100646	['Black Ivory']	0iijpRubQCC5cBUGd0U8sf	Mainline	1.0
...	...	...	...	...
50937	['Wolfgang Amadeus Mozart', 'Capella Istropoli...	7va2xRMFprSkJAjPvFipvF	Symphony No. 28 in C Major, K. 200: II. Andante	0.0
138448	['Shinedown']	5AtRcGbmFeHtnbQ4k93UW7	Second Chance	0.0
156053	['Sergei Rachmaninoff', 'William Kapell', 'Edm...	5g73emlmJYrMA25OL4lT1L	Cello Sonata in G Minor, Op. 19: III. Andante	0.0
51176	['Ludwig van Beethoven', 'Pinchas Zukerman', '...	6PvNbpPdHtKi4b2i3VXp17	Violin Romance No.1 In G Major, Op.40	0.0
51960	['George Gershwin']	2XSBXz4uDvx1PQPYJWQpcK	Rhapsody in Blue	0.0

170653 rows × 4 columns

Fig. 3. Database of Songs

In Fig 3 the list of songs order from highest probability to lowest probability is presented. After accomplishing this all that is needed is a bit of code that will eliminate duplicate songs that the user has already identify as liking, most likely songs that have a probability of 1.0. In addition, the code would obtain the ID presented on Fig 3 and search for it in the actual database that contains the song files. In order to accomplish this the design team would require access to Spotify's database and application environment, hence the project has reached completion.

V. Conclusion

The widespread use of the Internet has led to significant changes in the music industry regarding online music listening, copyright management, genre categorization, and music recommendations. The development of music recommendation systems is still ongoing, despite recent research showing the effectiveness of MIR techniques. Researchers have focused a lot of attention on automatically recommending music using audio properties, but even the best systems still fall short of expectations.

In light of this, a dataset of user-liked and -disliked songs was used in this study. The 14 acoustic characteristics of each song will be used to create a mathematical model that predicts the listener's musical preferences. To find the best model to suggest music to the user based on his preferences, the team investigated various machine learning and deep learning algorithms, including Decision Tree Classifier, KNN, Support Vector Machine, Random Forest Classifier, and a variety of deep learning architectures. After fine-tuning the hyper parameters of the algorithms and testing the different techniques, the team obtained an optimized Random Forest Classifier that yields an 83% accuracy in recommending songs that the user will like.

The team created a streamlined machine learning model that effectively predicts whether a user will like a song despite the difficulty of creating reliable and accurate music recommendation systems. Future research on this topic should examine how to deal with different variations of the same musical piece (such as cover songs), implement deep learning techniques to optimize the song's feature extraction process, and analyze multi-stakeholder music recommendations—considering that current music recommendation systems are frequently tuned to benefit a single stakeholder. In contrast, multi-stakeholder recommenders tend to be more objective, further improving the capabilities of MRSs.

Acknowledgment

We thank Dr. Hai Deng (Florida International University) for providing guidance on pattern recognition and its applications in the real wolrd.

References

[1] Peter J. Rentfrow and Samuel D. Gosling. The Do Re Mi's of Everyday Life: The structure and personality correlates of music preferences. Journal of Personality and Social Psychology, 84(6):1236–1256, 2003.

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

[3] Elbir, A., Bilal Cam, H., Emre Iyican, M., Ozturk, B., & Aydin, N. (2018). Music genre classification and recommendation by using Machine Learning Techniques. 2018 Innovations in Intelligent Systems and Applications Conference (ASYU). https://doi.org/10.1109/asyu.2018.8554016

[4] J-J Aucouturier and F. Pachet. Improving timbre similarity: How high's the sky. Journal of Negative Results in Speech and Audio Sciences, April 2004. 7

[5] A. Berenzweig, B. Logan, D.P.W. Ellis, and B. Whitman. A large-scale evaluation of acoustic and subjective music similarity measures. In Proceedings International Conference on Music Information Retrieval (ISMIR), 2003.

[6] Kim, D., Kim, K.-su, Park, K.-H., Lee, J.-H., & Lee, K. M. (2007). A music recommendation system with a dynamic K-means clustering algorithm. Sixth International Conference on Machine Learning and Applications (ICMLA 2007). https://doi.org/10.1109/icmla.2007.97

[7] Fessahaye, F., Perez, L., Zhan, T., Zhang, R., Fossier, C., Markarian, R., Chiu, C., Zhan, J., Gewali, L., & Oh, P. (2019). T-RECSYS: A novel music recommendation system using deep learning. 2019 IEEE International Conference on Consumer Electronics (ICCE). https://doi.org/10.1109/icce.2019.8662028

[8] Quasim, M. T., Alkhammash, E. H., Khan, M. A., & Hadjouni, M. (2021). Emotion-based music recommendation and classification using Machine Learning with IOT framework. Soft Computing, 25(18), 12249–12260. https://doi.org/10.1007/s00500-021-05898-9

[9] F. Pedregosa et al., "1.17. Neural network models (supervised)," scikit-learn. [Online]. Available: https://scikit-learn.org/stable/modules/neural_supervised.html. [Accessed: 01-Dec-2022].

[10] F. Pedregosa et al., "1.9. Naive Bayes," scikit-learn. [Online]. Available: https://scikit-learn.org/stable/modules/naive_bayes.html. [Accessed: 01-Dec-2022].

[11] F. Pedregosa et al., "1.6. Nearest neighbors," scikit-learn. [Online]. Available: https://scikit-learn.org/stable/modules/neighbors.html. [Accessed: 01-Dec-2022].

[12] F. Pedregosa et al., "1.10. decision trees," scikit. [Online]. Available: https://scikit-learn.org/stable/modules/tree.html. [Accessed: 01-Dec-2022].

[13] F. Pedregosa et al., "1.4. Support Vector Machines," scikit-learn. Available: https://scikit-learn.org/stable/modules/svm.html (Accessed: December 1, 2022).

[14] F. Pedregosa et al., "1.11. ensemble method," scikit-learn. Available at: https://scikit-learn.org/stable/modules/ensemble.html#ensemble (Accessed: December 1, 2022).