Proceedings of CONF-SEML 2025 Symposium: Machine Learning Theory and Applications
DOI: 10.54254/2755-2721/2025.TJ23297

Analysis of AI Evaluation for Music Pieces

Pinzhang Chen

Rutgers Preparatory School, Somerset, USA
pchen28@rutgersprep.org

Abstract: In recent years, an upsurge of Artificial Intelligence models and other machine learning methods has renovated the field of computer music, especially rating music using computers. A lot of methodologies are only based on numerical features of music to predict popularity scores, and the input parameters are often with limited information about the actual music. Moreover, they often focus on already existed features for published music. This research, on the other hand, introduces an improved method, which predicts the popularity score of raw audio file. The methodology includes a series of pre-trained audio feature extraction models, YAMNet and Librosa, to get audio feature data. The research also includes a RandomForest regression model that uses both numerical and categorical features to predict the popularity score of the input music. Tested results have shown a mean squared error (MSE) of 183.542 and a percentage error of 16.89%. This research emphasizes the possibility to advance music performance rating models as well as leading the direction for more well-developed advanced music rating applications.

Keywords: Music rating, YAMNet Pre-Trained Model, RandomForest regression, music statistics, Librosa.

1. Introduction

The field of computer music has evolved a lot since the invention of the first computers. Early research in computer music were started by researchers like Lejaren A. Hiller Jr. [1], who provided a first glimpse of computer music and provided a foundation for research in computer music. As the time goes on, in the 1970s, developments such as synthesizers and electronic instruments further boosted the field, enabling deeper research such as the overall application guide to computer music synthesis [2, 3] and signal processing of computer music [4].

Recently, as machine learning started to renovate the field of computer science, these techniques also provided plenty of applications to computer science, especially in topics such as classification of music genre [5], and music emotion recognition [6]. Some certain pre-trained models, like YAMNet and VGGish, emerged with well-developed features for music feature extraction, making further research, such as music rating, available. There are already plenty of research in music popularity based on AI, these studies have relied on using audio features data and determines the popularity [7], this study, on the other hand, aims to use popularity to represent rating score. These research treats music feature numerical data as the only input, while underseeing the potential audio characteristics that drive it. This article explores gathering popularity data by input the underlying audio features extracted from the raw audio file using music models like YAMNet and Librosa. Thus, making the rating of newly generated or composed music available by noticing the details in the raw audio.

DOI: 10.54254/2755-2721/2025.TJ23297

The motivation for this research is the need to connect the gap between predicting popularity based on music features and the extraction of music features from raw audio file using pre-trained models. This article aims to integrate traditional features to extractions from audio files. To be more specific, this research will include the following sections. The Sec. 2 will discuss dataset and models used in this research. An explanation of various datasets and models as well as the evaluation methods using these resources. Subsequently, this study will give procedures of model training, which covers a complete explanation of the procedures involved in training the prediction model. Subsequently, model performances and explanation are presented, including analysis of the result of the model. Afterwards, limitations and prospect are listed, which specifies the limitations in this method and prospect of improvements. Eventually, a conclude remark is given.

2. Data and Method

2.1. Dataset

In this research, a dataset from Kaggle is being used. The dataset was originally posted by the uploader as two separate datasets: a high popularity Spotify dataset with a popularity more than 68 out of 100, and a low popularity Spotify dataset that includes music with popularity lower than 68. After getting these datasets, they were merged into one dataset that provides comprehensive music that popularity is from 0 to 100. This dataset includes a wide variety of features, including numerical features as well as categorical features, making it a rich fit for the research of inputting rich features from the raw audio.

The numerical features include traditional features involved in predicting popularity, such as instrumentalness, loudness, acousticness, and liveness. Such features are already being used to detect the audio features of the music. Moreover, it includes statistics that value the overall features, like tempo, duration, energy, and time signature. These features provide a deeper understanding into analyzing the music. The categorical features include type, playlist genre, and playlist subgenre; these provided even more insight into the relationship between the popularity and the types of music. The dataset was posted in 2024, which includes music statistics up to November 2024. In addition, another important feature of it is that it corresponds to the extraction results of the pre-trained models YAMNet and Librosa, making the transition from extraction to prediction possible. Before training the model, a check into the dataset is performed for better accuracy, this includes dropping the rows with missing data and scaling each numerical feature to a range from 0 to 1, increasing accuracy by eliminating potential error by scaling differences.

2.2. Models and Evaluations

The audio extraction model is a series of pre-trained models, named YAMNet and Librosa, driven by AI, these two models can detect high-level details in music, such as timbre, rhythm, and BPM. After analyzing the audio, it outputs a 1024-dimensional vector graph providing its own summary, then, the graph can be converted to show the extracted features in the music. Another important part of this research is the popularity prediction model, it is built using a RandomForest regression module. The choice of RandomForest stands on its features of handling complex logics between features, and its mixed intelligence of including categorical features and combine it with numerical features. The audio is first processed using the audio extraction model, to get all the necessary audio features in the audio. Then, the extracted features serve as the input to the RandomForest model, which predicts its final popularity score. In terms of evaluation methods, the evaluation methods used in this research includes Root Mean Squared Error (RMSE), and the coefficient of determination (R^2). In addition, cross-validation is also applied to ensure the performance of the model. Last, the evaluation outputs a final percentage error for final reference, ensuring the accuracy of model.

DOI: 10.54254/2755-2721/2025.TJ23297

3. Results and Discussion

3.1. Model Training Procedure

In the process of music rating prediction, a RandomForest model is used. The music popularity dataset, which is combined from two matching datasets from Kaggle, is imported by the Python code. Then, the code was set to read each feature from the dataset, including numerical features and categorical features. After that, the missing values got handled, by dropping missing rows using Pandas, it produces a full and clean dataset with perfect data.

The next step in training the model is feature processing, which includes processing both numerical and categorical features. For numerical features, a process of removing the scaling difference is done, by using the MinMaxScaler, all numerical values, including popularity (from 0 to 100), or duration (in milliseconds), is all rescaled to a number within the range of 0 to 1. This process is crucial as it eliminates the possible accuracy impact by using un-scaled feature values. For categorical features, OneHotEncoder from sklearn.preprocessing is implemented to encode, which converts those categorical features to numerical values as most ML algorithms are not capable of working with direct categorical inputs. Following on, the next step is bringing these preprocessed features together. Two pipelines were used: numerical features pipeline, and categorical features pipeline. To train the model, a RandomForest Regression model is defined in the pipeline. Fig. 1 shows the overall structure of numerical features and categorical features.

The last step before training the model is dividing the dataset, in this research, the dataset is divided by 90% for training and 10% for testing. This proportion ensures that enough training data is received while leaving plenty of cases for testing. Finally, the model is being trained.

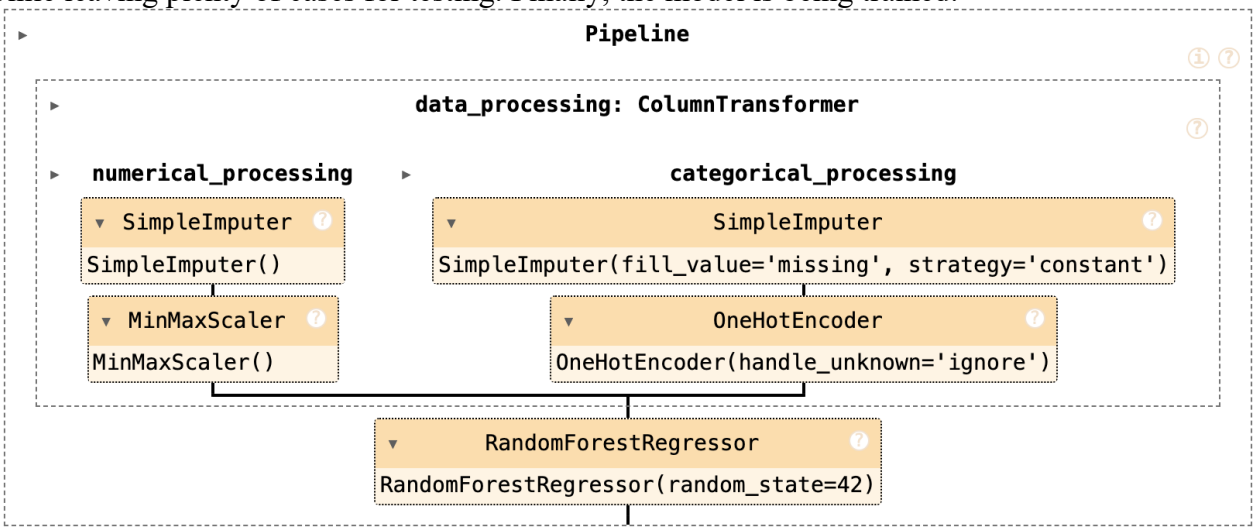

Figure 1: The structure of the RandomForest Regression model (Image generated using scikit-learn's set_config(display="diagram"); © scikit-learn developers).

3.2. Performances and Explanation

The performance of the outcome model is tested with the testing dataset, which is divided from the input Spotify dataset, using multiple ways. The model achieved a mean squared error (MSE) of 183.542, a r-squared error of 0.496, and a mean absolute error (MAE) of 9.732. The result of the r-squared error, 0.496, means that 49.6% percent of data can be explained by our model. The detailed table of each evaluation method and its results is shown in Table 1.

From training the RandomForest regression model, feature importance is also revealed in the process of training data. Fig. 2 shows the overall importance of each feature involved in the dataset

DOI: 10.54254/2755-2721/2025.TJ23297

input. Each bar represents a single feature, and the higher a feature means the more importance it has over determining the final popularity. As observed in the graph, most features with high importance scores are various forms of genres. This suggests that categorical types, which are genres and subgenres, plays an important role in popularity. In addition, other traditional numerical music features, such as duration, acousticness, valence, key, and tempo, proves to also be important in popularity value. This leads to a conclusion that both categorical and numerical features contribute to the popularity. On the other hand, features with lower importance are still a lot, meaning that the popularity decision is still driven by a relatively small number of features.

Table 1: The result of each evaluation method used.

Evaluation Method	Value
Mean Squared Error (MSE)	183.54215595238097
R-squared	0.4956802455410695
Mean Absolute Error (MAE)	9.731915113871636
Percentage Error	16.893135669362085

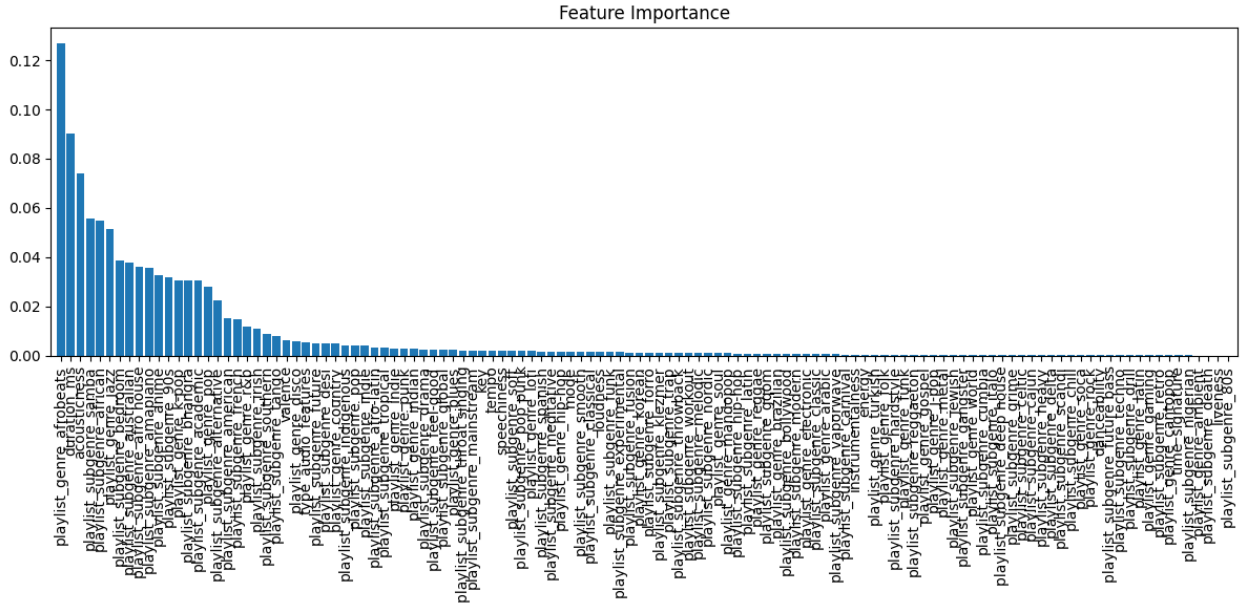

Figure 2: The feature importance of each feature used in training the RandomForest Regression model (Graph generated using Matplotlib; author-generated visualization).

3.3. Limitations and Prospect

Several limitations of this research were revealed during study. First, the use of popularity score as rating score has obvious drawbacks. Popularity scores often represent social trend, marketing success [8], etc. Popularity may not always purely represent the quality of the actual piece. For example, a song with average quality were popular over social media, while its popularity is high, its actual quality may not. Moreover, music with low popularity might also be with good quality [9]. During the training process, this limitation is also shown, as a lot of low popularity score below 10 is shown for a complete collection of classical symphonies composed by Beethoven. While his piece comes with great qualify, the model will still rate the pieces as low due to low popularity.

DOI: 10.54254/2755-2721/2025.TJ23297

Another drawback is from the dataset, as it is an overall statistical dataset fetched from Spotify API, the number of each genre is not equal, and large difference may be present [10]. This may result in the difference of performance for different genre of music.

Looking forward, future research could bypass these limitations in several improved ways. The dataset could be improved by collecting a pure rating-based dataset rather than a popularity dataset from rating websites. Moreover, datasets with not only ratings but expert comments could be provided, resulting in a rating and commenting model that helps the composer of the music by providing advice over improving the music. In terms of research methodologies, further research could try bypassing the transfer from audio file to features, and features to popularity by directly training a deep learning model using raw audio file with resulting rating score and comment. While it may require intensive data collection, it is possible.

In summary, while the model demonstrates the possibility and performance of rating music by extracting features and evaluating popularity, further research is suggested for increasing the consistency and accuracy of the model. This research could provide more information regarding rating of music.

4. Conclusion

To sum up, this research aims to provide a solution to music rating by extracting music features from music file, then predicting the popularity score as rating. For the methodologies of this research, YAMNet and Librosa pre-trained model for extracting music features and a RandomForest regression model for processing popularity is chosen. The research results have shown that rating music through this method is effective, with the model achieving a mean squared error (MSE) of 183.542 and a percentage error of 16.89%, which proves that rating music through popularity is possible. Looking forward, further experiments could focus on incorporating music statistics from multiple sources, balancing the number of pieces in each different genres, featuring a pure rating dataset instead of using popularity as rating scores. Advanced studies could also focus on training a model by providing direct raw audio data with rating score and comments. This research emphasizes the possibility to rate raw audio file such methodologies, provides inspiration regarding music rating, and gives potential application of raw music rating systems.

References

[1] *Hiller, L.A. (1959) Computer music. Scientific American, 201(6), 109–121.*
[2] *Dodge, C., Jerse, T.A. (1985) Computer music: Synthesis, composition, and performance (1st ed.). Macmillan Library Reference.*
[3] *Loy, G., Abbott, C. (1985) Programming languages for computer music synthesis, performance, and composition. ACM Computing Surveys (CSUR), 17(2), 235-265.*
[4] *Moorer, J. A. (1977) Signal processing aspects of computer music: A survey. Proceedings of the IEEE, 65(8), 1108–1137.*
[5] *Pelchat, N., & Gelowitz, C. M. (2020). Neural network music genre classification. Canadian Journal of Electrical and Computer Engineering, 43(3), 170-173.*
[6] *Kim, Y.E., Schmidt, E.M., Migneco, R., et al. (2010) Music emotion recognition: A state of the art review. In Proc. Ismir, 86, 937-952.*
[7] *Lee, J., Lee, J.S. (2018) Music popularity: Metrics, characteristics, and audio-based prediction. IEEE Transactions on Multimedia, 20(11), 3173-3182.*
[8] *O'Reilly, D. (2007) The marketing of popular music. In Arts Marketing (pp. 6-25). Routledge.*
[9] *Parakilas, J. (1984) Classical Music as Popular Music. The Journal of Musicology, 3(1), 1–18.*
[10] *Murphey, Y.L., Guo, H., Feldkamp, L.A. (2004) Neural learning from unbalanced data. Applied Intelligence, 21(2), 117-128.*