A Comparative Study of Decision Tree and Random Forest Regression in Predicting Song Popularity

Rainnand Montaniel
College of Computing and Information Technology
National University
Manila, Philippines
montanielrp@students.national-u.edu.ph

Karl Satchi Navida
College of Computing and Information Technology
National University
Manila, Philippines
navidake@students.national-u.edu.ph

Renaire Odarve
College of Computing and Information Technology
National University
Manila, Philippines
odarverb@students.national-u.edu.ph

***Abstract*—This research aims to determine the development of the song popularity prediction based on the audio features from Spotify. Three feature extraction techniques are explored: Standard Fractional Undersampling, Multiple Fractional Undersampling, and Recursive Feature Elimination (RFE). Feature sets are fed to two algorithms, Decision Tree Regression (DTR) and Random Forest Regression (RFR). The results were then compared to see which algorithm can produce the best results in predicting the popularity of the song. Experiments show that RFR outperforms DTR. Specifically, RFR with Multiple Fractional Undersampling achieved the most accurate prediction with an RMSE of 25.90 and MAE of 22.81.**

***Index Terms*—machine learning, song popularity, Spotify, audio features, Recursive Feature Elimination, Undersampling, Random Forest Regression, Decision Tree Regression.**

I. Introduction

Understanding the characteristics that contribute to a song's popularity has become increasingly data-driven in today's era. Audio features such as key, pitch, and acoustic level are now being used as benchmarks to create machine learning models, for high-level descriptive tasks like music tagging, genre classification, and key detection [1]. The development of all these technological advancements in music and data collection has led us to understand more nuances about the concept of popularity. Spotify, —one of the music platforms that is spearheading the digital age consumption of music— offers a vast collection of data music from its 626 millions of users and 246 million subscribers with over 40,000 new tracks uploaded daily, amounting to millions annually [2]. However, only a small fraction of the tracks achieves significant popularity.

This research aims to assess the viability of Spotify's audio features as predictors of song popularity, using select machine learning models. By using these models, the research provides significant data-driven analytics and comparison on the effectiveness of different algorithms in predicting song popularity.

II. Review of Related Works

Popular music is defined as a "recorded musical composition that is loved, well-known, admired and enjoyed by the general public" [3]. The public, for the study's purposes, use Spotify at least once to listen to music. Audio features are defined as a specific property of an audio signal, or in this case of the study, a track [4]. Examples of audio features include loudness, acousticness, energy, danceability, etc. These features enable the development of advanced algorithms that enhance music quality and improve user experiences on various music platforms [5].

Examples of implementation include personalized recommendation systems and genre classification [5]. With advancements in technology that reshaped mainstream music consumption, it is possible that audio features significantly impact the song's popularity [3].

Existing works include a study conducting exploratory data analysis on the audio features of K-pop to understand its global popularity despite cultural and language barriers [6]. The study concluded that K-pop music has elevated levels of danceability, average levels of energy and cheerfulness, and a comprehensible level of speech or lyrics in the song. However, machine learning models were not implemented in the study [6].

Another study uses different musical features and deep neural networks for popularity prediction [7]. While the research achieved success, the datasets that were used are a combination of descriptive (e.g., genres, moods, vocal) and numerical (e.g., valence, energy, key) features, leading to classification imbalance in terms of popularity [7]. This imbalance occurred because the dataset was more biased towards western vocal music [7].

This research focuses more on the numerical audio features of a music as a predicting factor of popularity level, aiming to avoid classification imbalance that may adversely affect the accuracy of the predictive analysis.

III. Methodology

The overview of the processes made for song popularity prediction is shown in Figure 1. The development is divided into four parts: data collection, data pre-processing, modelling, and evaluation. Data collection is where the process of collecting the data happens. Data pre-processing is where the collected data is cleaned and formatted using Undersampling and Recursive Feature Elimination to be ready for the next step. In the Modelling stage, the processed data will then be used by two machine learning models, Decision Tree Regression and Random Forest Regression to learn the patterns. The last step is the Evaluation, where preset test data will be used on each of the generated models to check their accuracy to then use for predicting the popularity of the songs.

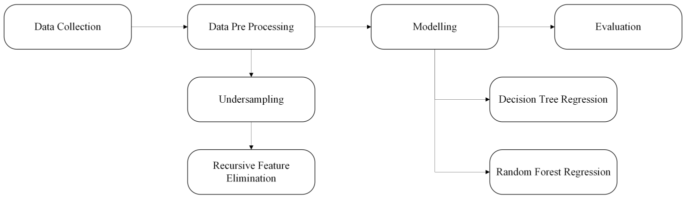

Fig. 1. Song Popularity Prediction Development Framework

A. Data Processing

Three datasets were collected for this research. The first dataset, obtained from the Kaggle open-source database collection, contains 23 columns and 30,000 rows of music data [8]. This dataset was selected for its detailed representation of audio features, which supports accurate popularity prediction, making it suitable as a training dataset.

The second dataset was also retrieved from Kaggle. It is labeled as ”Standard Tracks” dataset with 114,000 observations. This dataset provides a broad range of tracks for assessing model performance for music popularity prediction, and it will also be used as training data [9]. The popularity level distribution is shown in Figure 2, with 20,769 observations having popularity levels between 0 and 5 out of a total of 114,000 songs.

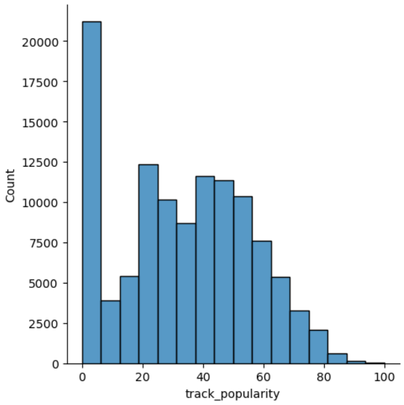

Fig. 2. Standard Tracks Dataset Popularity Distribution

The third dataset was sourced from HuggingFace. It is a ”Top Tracks” dataset, containing 1,823 highly popular songs out of 2,000 observations, making it appropriate as training data [10]. Figure 3 shows the popularity distribution, where only 177 observations of music data having popularity levels between 0 and 5.

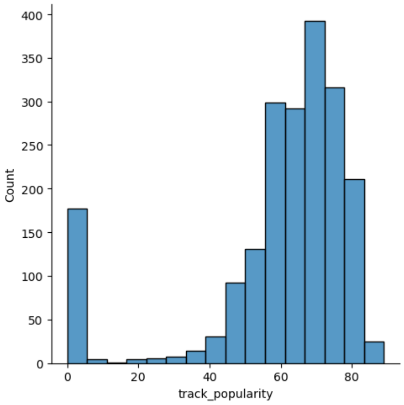

Fig. 3. Top Tracks Dataset Popularity Distribution

B. Data Pre-Processing

The following pre-processing techniques were applied to prepare the data for popularity prediction:

1) Data cleaning: Incomplete, duplicate, and null values were excluded from the three datasets. Missing values were also excluded to ensure the quality of training and testing data. Numerical audio features were extracted, which are danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, and duration

2) Undersampling: To address classification imbalance that might occur in popularity levels, undersampling techniques were applied to the training data. The purpose of undersampling is to balance the data by reducing the number of overrepresented samples to achieve similar counts across among other classes. Undersampling is an efficient method for handling imbalanced data, as it reduces the processing load by ignoring portions of the majority class in the observations. However, one drawback is that useful information might be lost within the excluded samples [14].

This research employs two types of undersampling: standard fractional and multiple fractional undersampling.

a) Standard fractional undersampling: This approach retains only 10% of samples with a 0-valued popularity level. See Figure 4.

b) Multiple fractional undersampling: This approach targets popularity levels between 0 and 5, retaining only 40% of samples within this range. See Figure 5.

To address the potential loss of useful information from undersampling, this study includes an additional training dataset with no undersampling applied as a comparison for predictive accuracy (see Figure 6). These approaches will be applied to the training data for comparison across two different machine learning models.

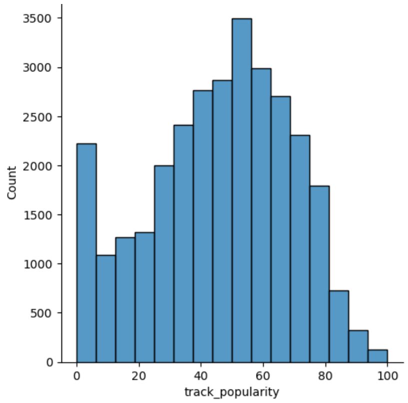

Fig. 4. Popularity Distribution with Standard Fractional Undersampling

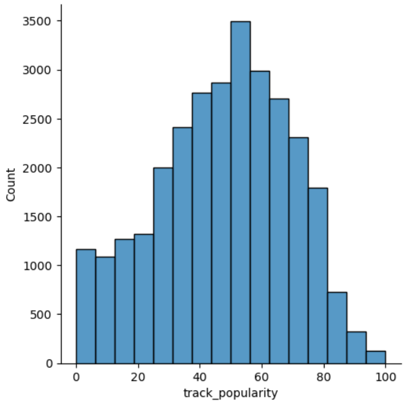

Fig. 5. Popularity Distribution with Multiple Fractional Undersampling

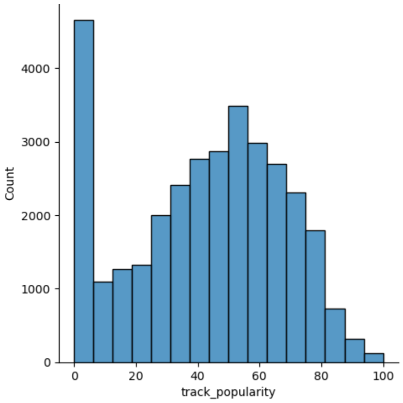

Fig. 6. No Undersampling Training Data Popularity Distribution

3) Recursive Feature Elimination (RFE): RFE was applied to retain only the key features that contribute most to prediction accuracy. It works by searching for a subset of features by starting with all features in the training dataset and successfully removing features until the desired number remains. This is achieved by fitting the given machine learning algorithm used in the core of the model, ranking features by importance, discarding the least important features, and refitting the model. This process is repeated until a specified number of features remains [15].

C. Experimental Setup

This research utilized Python (Jupyter Notebook in Visual Studio Code) with libraries such as Numpy, Pandas, Matplotlib, Seaborn, and Scikit-Learn. Pandas facilitated data manipulation, Seaborn and Matplotlib were used for dataset visualization, and Scikit-Learn provides essential tools for model training and evaluation, including standard scaler, cross-validation, and error metrics.

The study involves two experimentation settings that would be conducted, namely, the ”Standard Tracks” and ”Top Tracks” training dataset. These two training data would be the basis for the machine learning algorithms that would be used for this study. Additionally, further comparisons would be analyzed between the three types of undersampling data based on their predictive accuracy between the two experimentation settings.

D. Algorithms

The algorithms used in this research for predicting song popularity are the Decision Tree Regression and Random Forest Regression.

1) Decision Tree Regression (DTR): As the name implies, Decision Tree follows a tree-like structure of decisions. It observes a set of data and trains a model to predict future data based on its observations. It can be further classified into two methods, classifier and regression. The main difference between the two is that classifier is used when the output is discrete (predicting weather) while regression is when the output is continuous (predicting profit, popularity) [11].

2) Random Forest Regression (RFR): Random Forest Regression uses multiple decision trees as learning base models and tries to combine the prediction of all of them. The base models are taken from a subset from the data are learning parallel to each other. This technique is called Bootstrap Aggregating, or Bagging for short. It decreases the variance and avoids overfitting [12].

E. Training Procedure

The training process utilized cross-validation with KFold from Scikit Learn model selection module to assess model performance reliably. Kfold splits the data into k subsets and trains the model on k minus one folds and validates it on the remaining fold, repeating this process k times to reduce variance and improve model accuracy. [13]

F. Evaluation Metrics

The study employs both Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), two types of metrics to evaluate model performances of both Random Forest regression and Decision Tree Regression based on the two types of testing data set.

1) Root Mean Squared Error (RMSE): Root Mean Squared Error (RMSE) measures the average difference between predicted outcomes and observed values, [16]. as shown in Figure 7 below.

$$(y, \hat{y}) = \sqrt{\frac{\sum_{i=0}^{N-1}(y_i - \hat{y}_i)^2}{N}}$$

Fig. 7. RMSE Formula

Where

- $\sum$ represents the summation symbol
- N represents the size of the dataset
- y_i predictive value of the ith observation
- $\hat{y}_i$ actual value of the ith observation

2) Mean Absolute Error (MAE): MAE calculates the average absolute difference between predicted and observed values, a straightforward calculation is shown on Figure 8 below.

$$(y, \hat{y}) = \frac{\sum_{i=0}^{N-1} |y_i - \hat{y}_i|}{N}$$

Fig. 8. MAE Formula

Where

- $\sum$ represents the summation symbol
- N represents the size of the dataset
- y_i predictive value of the ith observation
- $\hat{y}_i$ actual value of the ith observation

IV. Results and Discussion

In this study, the performance of the Random Forest Regression and Decision Tree Regression is being evaluated based on the following combinations of undersampling techniques:

- **RFR-N**: Random Forest Regression with no undersampling technique
- **RFR-M**: Random Forest Regression with multiple fractional undersampling technique
- **RFR-S**: Random Forest Regression with standard fractional undersampling technique
- **DTR-N**: Decision Tree Regression with no undersampling technique
- **DTR-M**: Decision Tree Regression with multiple fractional undersampling technique
- **DTR-S**: Decision Tree Regression with standard fractional undersampling technique

Different categories were created to assess how the audio features of a track predict its popularity level. Since classification imbalance can affect predictions, undersampling techniques were applied to assess whether rebalancing the data impacts predictive accuracy.

This study proposes to conduct two experiments on both the Standard and Top Tracks datasets to determine the accuracy of the model as outlined in the experimental setup. Experiment 1 used the standard tracks with 18.22% (20,769) of low popularity level (0-5) out of 114,000 tracks. Experiment 2 are top tracks consisting of 9% (177) low level popularity tracks out of 2000 tracks.

TABLE I
Predictive Accuracy for the First Test Data

Accuracy	**RMSE**	**MAE**
RFR-N (No undersampling)	24.30	19.73
RFR-M (Multiple fractional undersampling)	26.00	20.99
RFR-S (Standard fractional undersampling)	25.34	20.51
DTR-N (No undersampling)	32.97	26.75
DTR-M (Multiple fractional undersampling)	31.79	25.76
DTR-S (Standard fractional undersampling)	32.28	26.14

Table 1 shows the predictive accuracy for Standard Tracks dataset. Among the approaches, RFR-N (Random Forest Regression without undersampling) demonstrated the lowest RMSE of 24.30 and MAE of 19.73, indicating a more accurate prediction when no undersampling technique was applied. The RFR-M and RFR-S categories had slightly higher RMSE values (26.00 and 25.34, respectively), suggesting that undersampling might reduce accuracy when predicting datasets with a large volume of low-popularity tracks. For Decision Tree Regression, DTR-M (Decision Tree Regression with multiple fractional undersampling) yielded the lowest error rates (RMSE 31.79 and MAE 25.76), suggesting a slight advantage over the standard and no-undersampling techniques in this model.

TABLE II
Predictive Accuracy for the Second Test Data

Accuracy	**RMSE**	**MAE**
RFR-N (No undersampling)	28.95	26.53
RFR-M (Multiple fractional undersampling)	25.90	22.81
RFR-S (Standard fractional undersampling)	26.99	24.13
DTR-N (No undersampling)	37.63	30.13
DTR-M (Multiple fractional undersampling)	38.77	26.34
DTR-S (Standard fractional undersampling)	34.15	27.48

Table 2 displays the predictive accuracy for the Top Tracks dataset, where tracks with popularity levels greater than 5 comprise 91.15% (1,823) of the data. The RFR-M (Random Forest Regression with multiple fractional undersampling) approach achieved the lowest RMSE of 25.90 and MAE of 22.81, indicating that undersampling techniques provide the most accurate predictions for datasets with a high volume of popular tracks. RFR-S followed closely with an RMSE of 26.99 and MAE of 24.13, while RFR-N and DTR-N yielded the highest error rates within their respective model categories. These findings suggest that models without undersampling are less effective when dealing with datasets dominated by higher popularity tracks.

The results for both the Standard Tracks and Top Tracks datasets highlight differences in model accuracy due to dataset volume and distribution. In the Standard Tracks dataset, RFR-N achieved the lowest RMSE and MAE, while in the Top Tracks dataset, RFR-M yielded the lowest RMSE and MAE. Overall, Random Forest Regression outperformed Decision Tree Regression across both datasets, suggesting that Decision Tree Regression is more susceptible to imbalanced class distributions, as reflected in its higher error rates. In contrast, Random Forest Regression maintained reliable performance across all sampling techniques applied.

V. Conclusion

This paper has explored the effectiveness of Random Forest Regression (RFR) and Decision Tree Regression (DTR) with applied sampling techniques, to predict the popularity of Spotify tracks based on audio features. Data cleaning and Recursive Feature Elimination (RFE) were implemented, as well as the utilization of different undersampling techniques to address the imbalance distribution of the training dataset. The predictive capability of the models was evaluated using Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) metrics

The experimental results indicate that Random Forest Regression outperformed Decision Tree Regression in both datasets. Specifically, RFR-N achieved the lowest RMSE and MAE in the Standard Tracks dataset, suggesting its effectiveness for widely distributed popularity levels. Similarly, RFR-M achieved the most accurate prediction among all tested models in the Top Tracks dataset. This indicates that RFR performed better generally on predicting the popularity levels of Spotify tracks using its audio features.

Given that the focus of this study was the numerical features, the study acknowledges potential improvements of predictive capability through utilizing additional features, such as artist, lyrics, album, genre, and sub-genre. However, these features are descriptive and would additionally require extensive Spotify API support for feature generation and advanced undersampling and encoding techniques for dataset optimization. Future studies could also benefit from exploring advanced machine learning approaches like sequential learning to capture complex patterns in track popularity, which is scalable enough to be utilized for predictive analysis and potentially, music recommendation systems [13].

References

[1] R. Yuan, Y. Ma, Y. Li, G. Zhang, et al. (10-19 Dec. 2023). “MARBLE: Music Audio Representation Benchmark for Universal Evaluation,” in NeurIPS 2023 Datasets and Benchmarks Track, New Orleans, USA, [Online]. Available: https://proceedings.neurips.cc/paper\textunderscorefiles/paper/2023/file/7cbeec46f979618beafb4f46d8f39f36-Paper-Datasets_and_Benchmarks.pdf.

[2] C. Williams, “Music Streaming Hits Major Milestone as 100,000 Songs are Uploaded Daily to Spotify and Other DSPs”. Variety. https://variety.com/2022/music/news/new-songs-100000-being-released-every-day-dsps-1235395788/ (accessed Oct. 30, 2024).

[3] H. S. Saragih, “Predicting song popularity based on Spotify’s audio features: insights from the Indonesian streaming users”, Journal of Management Analytics. vol. 10, issue 4, pp 693-709. July 2023. Accessed: Oct. 30, 2024. doi: 10.1080/23270012.2023.2239824. [Online] Available: https://www.tandfonline.com/doi/full/10.1080/23270012.2023.2239824.

[4] D. Mitrovic, M. Zeppelzauer, and C. Breitenede, “Features for content-based audio retrieval” in Advances in Computers. 1st Ed., Vol. 78, M. Zelkowitz, Massachusetts, USA: Academic Press, 2010, ch. 3, pp. 71–150.

[5] D. Sijbesma, “The Impact of Audio Features on Music Genre Classification and Recommendations,” M. S. Thesis, ADE, Utrecht Univ., Netherlands, 2024. [Online]. Available: https://studenttheses.uu.nl/bitstream/handle/20.500.12932/47974/Thesis%20-%20ADE%20-%20David%20Sijbesma.pdf?sequence=1.

[6] S. Miroudot, “What’s behind the ‘K’? Common audio features of Korean popular music before and after the rise of K-POP,” Popular Music, pp. 1–22, 2024. doi:10.1017/S0261143024000187

[7] E. Zangerle, M. Vötter, R. Huber, and Y.-H. Yang, “Hit Song Prediction: Leveraging Low- and High-Level Audio Features.,” in ISMIR, 2019, pp. 319–326 [Online]. Available: http://dblp.uni-trier.de/db/conf/ismir/ismir2019.html#ZangerleVHY19

[8] C. Thompson, J. Parry, D. Phipps, and T. Wolff, 2020, “Spotify Songs”, Spotify. [Online]. Available: https://github.com/rfordatascience/tidytuesday/blob/master/data/2020/2020-01-21/readme.md

[9] M. Pandya, 2022, “Spotify Tracks Dataset”, Spotify. [Online]. Available: https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset.

[10] O. Sanseviero, 2022, “Top Hits Spotify”, Spotify, [Online]. Available: https://huggingface.co/datasets/osanseviero/top-hits-spotify

[11] GeeksforGeeks, “Python — Decision Tree Regression using sklearn,” GeeksforGeeks, Jan. 11, 2023. https://www.geeksforgeeks.org/python-decision-tree-regression-using-sklearn/ (accessed Nov. 1, 2024)

[12] GeeksforGeeks, “Random Forest regression in Python,” GeeksforGeeks, Sep. 04, 2024. https://www.geeksforgeeks.org/random-forest-regression-in-python/ (accessed Nov. 1, 2024)

[13] Yin, S., Ruan, X., Song, J., and Zheng, W. (2024). Music Recommendation Algorithm based on Dual-Stream Sequence Fusion. In Lecture notes in computer science (pp. 328–339). https://doi.org/10.1007/978-981-97-5663-6_28

[14] N. X.-Y. Liu, N. J. Wu, and N. Z.-H. Zhou, “Exploratory Undersampling for Class-Imbalance Learning,” IEEE Transactions on Systems Man and Cybernetics Part B (Cybernetics), vol. 39, no. 2, pp. 539–550, Dec. 2008, doi: 10.1109/tsmcb.2008.2007853

[15] “RFE,” Scikit-learn.https://scikit-learn.org/dev/modules/generated/sklearn.feature_selection.RFE.html (accessed Nov. 1, 2024)

[16] “Root Mean Squared Error (RMSE) — SAP Help Portal.” https://help.sap.com/docs/SAP_PREDICTIVE_ANALYTICS/41d1a6d4e7574e32b815f1cc87c00f42/5e5198fd4afe4ae5b48fefe0d3161810.html