Journal Collabits
Vol. 3 No. 1 January 2026 : 1-8
E-ISSN : 3046-6709, P-ISSN : 3062-8601
https://publikasi.mercubuana.ac.id/index.php/collabits

Comparison of Random Forest and Naive Bayes Algorithms in Classification of Song Popularity on the Spotify Platform

Janu Ilham Saputro[1*], Rian Fantomi[2], Saoloan Simbolon[3], Puput Nur Arizka[4], Berlina Ramadani[5]

[1, 2, 3, 4, 5] Information System, Universitas Raharja, Indonesia

*Coressponden Author: janu@raharja.info

Abstract—The purpose of this study is to use machine learning to rank Spotify songs based on how popular they are. Because there is so much music data out there, musicians and artists need to know if a song will be popular or not. The dataset has 8,778 songs, each with different features like how popular the artist is, how many followers they have, and other song details. This research evaluates the efficacy of two classification algorithms: Random Forest and Naive Bayes. Artist popularity, artist followers, explicit album total tracks, and track number are the main things that are used to make models. The results of the experiment show that the Random Forest algorithm works better than the Naive Bayes algorithm. The Random Forest algorithm was right 76.54% of the time, but the Naive Bayes algorithm was only right 72.21% of the time. The f1-score for both popularity classes is also better for Random Forest. This finding shows that ensemble-based models, like Random Forest, work better with the features of music popularity data than basic probabilistic models do.

Keywords :
Spotify;
Random Forest;
Naive Bayes;
Classification Genre;
Machine Learning;

Article History:
Received: 15-11-2025
Revised: 25-12-2025
Accepted: 06-01-2026

Article DOI : 10.22441/collabits.v3i1.37578

INTRODUCTION

The music scene around the world is very different now than it was ten years ago, mostly because of the popularity of services like Spotify. These services give you access to an infinite library of music, but more importantly, they collect a huge amount of data about what users listen to and why. If you are serious about the business side of music, whether you work for a label or are an independent artist, you need to know the hidden patterns behind a song's rise to fame in order to create marketing campaigns that work.

It is still hard to figure out whether a song will be a hit or a miss based on data from before it comes out. This study suggests a remedy through the utilization of machine learning methodologies to develop a binary classification system. We want to use data mining to objectively divide songs into popular and unpopular tiers, using features that are available before the song even makes it to the top of the charts. This way, we won't have to rely on our gut feelings.

We are comparing two specific algorithms in terms of how they work. On one side is Random Forest, which was chosen because it can handle messy, non-linear data well.

On the other side is Naive Bayes, which was chosen because it is so fast. It's clear that this work needs to be done quickly: the music industry moves quickly, and producers need reliable predictive models that can help them make better, data-driven decisions.

LITERATURE REVIEW

In the last ten years, machine learning has become a lot better at working with music datasets. A lot of the research that has already been done has looked at how to use intrinsic audio attributes like tempo or loudness to sort songs by genre or mood. These features are great for putting together similar tracks, but they don't always work well for predicting the market. To get around this problem, more and more research is focusing on high-level metadata, which says that information about the artist and album is a better way to measure popularity than just the sound of the music.

We chose Random Forest and Naive Bayes as the algorithms to use for this analysis. We chose Random Forest because it can handle complicated, non-linear data by combining many decision trees. This makes it much less likely to overfit. Naive Bayes, on the other hand, is a probabilistic counterpoint. Even though it "naively" assumes that features are independent, it is still a standard baseline for classification tasks because it is easy to use and efficient.

This study is unique because it shifts its focus from audio signal processing to a dataset that is full of metadata. We examine the impact of "star power" by evaluating the correlation between an artist's established metrics—such as follower count and popularity score—and the commercial success of their releases. The goal is to find out if a song's success is mostly due to the quality of the sound or the artist's existing social capital.

METHODOLOGY

1.1 Data Acquisition and Scope

The empirical basis of this study is a specific dataset from the Spotify platform, saved as track_data_final.csv. There are 8,778 records in this dataset, which is a complete collection of music metadata. The data is organized into 15 different columns, each of which holds a different type of information. For example, track_id and track_name are unique identifiers, and there is also more general information about the performers. There is enough data here to look for patterns and check how reliable the classification models are.

1.2 Feature Selection Strategy

A strict process for selecting features was used to make sure that the machine learning models worked well. This study does not use all of the raw data that is available. Instead, it focuses on certain variables that are thought to have the most predictive power when it comes to how well a song will do in the market. The input vector for the classification task has the following features:

- artist_popularity: A number that shows how popular the artist is right now.
- artist_followers: The total number of people who subscribe to the artist.
- explicit: A flag that tells you if there is explicit content.
- album_total_tracks: The number of songs on the album that the track is on.
- track_number: The exact place of the song on the album

1.3 Target Variable Discretization

The main goal of this study is to guess how popular a song will be on the radio. But since the raw popularity score is often a continuous number, it was necessary to change this variable so that it could be used in a binary classification problem. The target variable was divided into two separate groups:

- **Class 0 (Unpopular):** This means that the tracks are not popular enough to meet the set popularity level.
- **Class 1 (Popular):** This means that the tracks meet or exceed the standards for success.

1.4 Class Distribution

The balance of the dataset is an important part of any classification task. When we look at the transformed target variable, we see that the dataset has a very good distribution: 4,442 examples are labeled as Class 1 (Popular) and 4,336 examples are labeled as Class 0 (Unpopular). This near-perfect balance is important for the integrity of the research because it lowers the chance that the model will become biased toward the majority class. This is a common problem in data mining tasks where one category is much larger than the other.

1.5 Algorithmic Framework

- **Random Forest:** This model is set up to use ensemble learning. During the training phase, it builds a "forest" of several decision trees. The final prediction is based on the votes of the majority of the trees. This method was chosen because it is strong and can lower the variance that comes with using only one decision tree.
- **Naive Bayes:** This classifier uses Bayes' Theorem to set a probabilistic baseline. It works on the strong assumption that having one feature doesn't affect having other features. Even though this makes things easier, it is still included to see if a model that is easy to compute can compete with more complicated ensemble methods.

1.6 Performance Evaluation Protocols

To rigorously assess the efficacy of the proposed models, the evaluation is not limited to a single metric. Instead, a comprehensive suite of indicators is employed to provide a holistic view of performance:

- **Accuracy:** Measures the overall percentage of correct predictions.
- **Precision:** Evaluates the exactness of the model (positive predictive value).
- **Recall:** Measures the completeness of the model (sensitivity).

F1-Score: Provides a harmonic mean between Precision and Recall, offering a balanced view of the model's effectiveness, particularly in distinguishing between the two classes.

RESULTS AND DISCUSSION

A. Exploratory Analysis:

Correlation of characteristics in exploratory analysis We did a full correlation analysis to see how the independent variables (features) and the target variable, track_popularity, were related before we started training the model. This statistical data is very important because it shows that the model's features can really tell the difference between a hit song and one that isn't.

Figure 1. Correlation Heatmap of Feature Variables

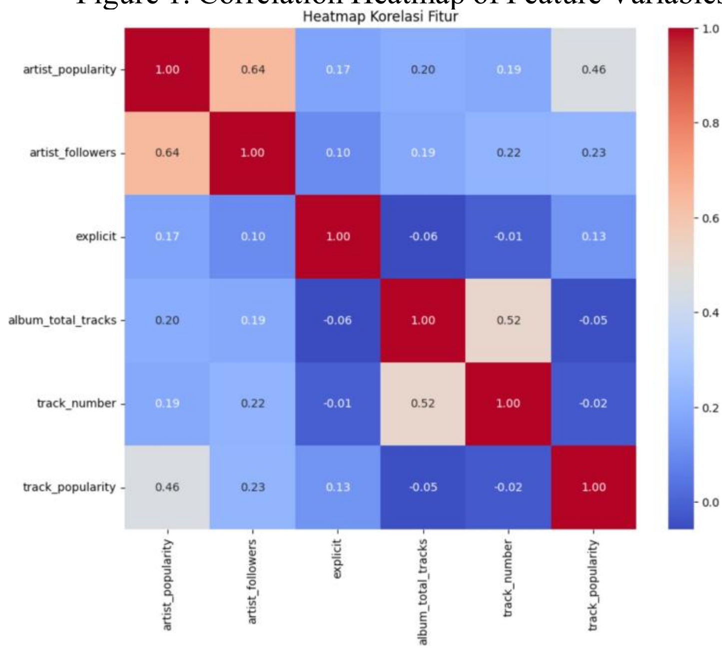


Figure 1 (Correlation Heatmap) shows that the data has different patterns. The feature artist_popularity is most closely related to how well the song does. This trend shows a clear "momentum

effect": songs by artists who are already famous are much more likely to become popular right away. Artist_followers also has a strong positive correlation, which backs up the idea that having a lot of fans before streaming is important for success in the streaming age. Explicit content and some album features, on the other hand, don't seem to be as closely related. But they are still helpful because they make the model more interesting, which helps it find patterns that aren't as clear.

B. Performance Analysis: Naive Bayes

We used a normal 70:30 split between training and testing to make sure our method worked. We used Accuracy, Precision, Recall, and F1-Score as a set of metrics to check how well things were working. This way, we could see the whole picture, since accuracy alone can be misleading when it comes to classifying things.

i. The Random Forest's results to make the Random Forest model as stable as possible, it was made up of 300 decision trees. The test set did very well, with an overall accuracy of 76.54%. Here is a detailed look at how well it can classify.

Table 4.1. Random Forest Classification

Class	Precision	Recall	F1-Score	Support
0 (Not Popular)	0.81	0.68	0.74	1301
1 (Popular)	0.73	0.85	0.79	1333
Accuracy			**0.77**	**2634**
Macro Avg	0.77	0.76	0.76	2634
Weighted Avg	0.77	0.77	0.76	2634

Table 4.1 shows that the Random Forest model has a very high Recall for Class 1 (0.85). This means that the model is very sensitive to hits. It finds most popular songs with a low rate of false negatives, so it's a good way to find songs that could make the charts.

ii. Outcomes from Naive Bayes We standardized the feature set and then used the Naive Bayes (GaussianNB) algorithm to compare. The model got 72.21% of the answers right, which is a little less than the ensemble method.

Table 4.2. Naive Bayes Classification

Class	Precision	Recall	F1-Score	Support
0 (Not Popular)	0.79	0.60	0.68	1301
1 (Popular)	0.68	0.84	0.75	1333
Accuracy			**0.72**	**2634**
Macro Avg	0.73	0.72	0.72	2634
Weighted Avg	0.73	0.72	0.72	2634

Naive Bayes can find popular songs, just like Random Forest can (recall 0.84), but its precision for Class 1 drops to 0.68. This means that the model is a little too "optimistic" because it often incorrectly classifies songs that aren't popular as popular (false positives) because it doesn't have the nuance to filter out the noise well.

Table 4.3. Model Performance Comparison

Algorithm	Accuracy	Precision (Avg)	Recall (Avg)	F1-Score (Avg)
Random Forest	**76.54%**	0.77	0.76	0.76
Naive Bayes	72.21%	0.73	0.72	0.72

iii. Comparative Analysis

Figure 4. Accuracy Comparison: Random Forest vs. Naive Bayes

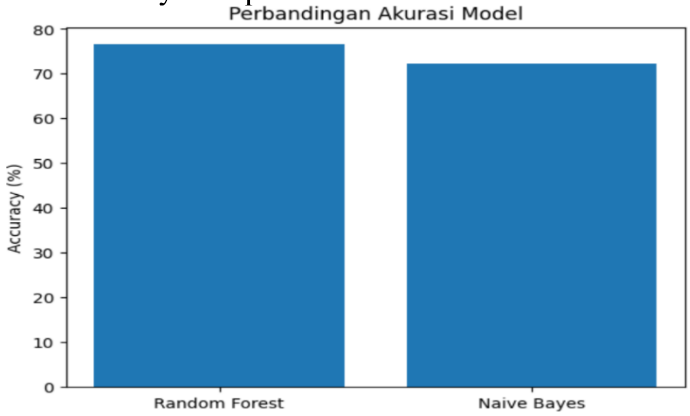
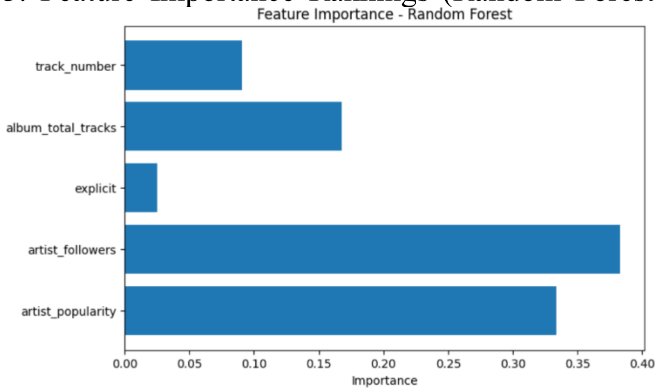


Figure 4 shows a direct head-to-head comparison that makes it clear how much better one method is than the other. It is clear that Random Forest is about 4.3% better than Naive Bayes. The main reason for this superiority is that the algorithms are built differently. Naive Bayes assumes that all features are separate from one another. When looking at music data, this is a problem because "artist popularity" and "follower count" are naturally linked. But this level of complexity is fine for Random Forest. It works well as an ensemble method because it captures how these features interact with each other in a non-linear way, which makes predictions that are both more accurate and more stable.

iv. Key Drivers of Success: Feature Importance

To provide actionable intelligence for the music industry, we extracted the Feature Importance scores from the best-performing model (Random Forest). The goal was to answer a critical question: *What actually drives a song's popularity?* The results are visualized in figure 3 below.

Figure 3. Feature Importance Rankings (Random Forest Model)



The data speaks clearly: artist_popularity ranks as the single most dominant factor, followed closely by artist_followers. This finding empirically confirms the "Star Power" hypothesis within the Spotify ecosystem. The success of a track is less about the audio characteristics of the song itself and more about the brand power of the artist. Established artists with massive followings benefit from stronger algorithmic recommendations, allowing their releases to reach a wider audience instantly. While features like explicit content and track_number play minor roles, they are secondary to the overwhelming influence of the artist's established reputation.

We used artist_popularity and artist_followers, the two most important features, to make a scatter plot that shows these results. Figure 4 shows how the data is spread out.

Figure 4. A scatter plot that shows how the number of followers and artist popularity are all connected

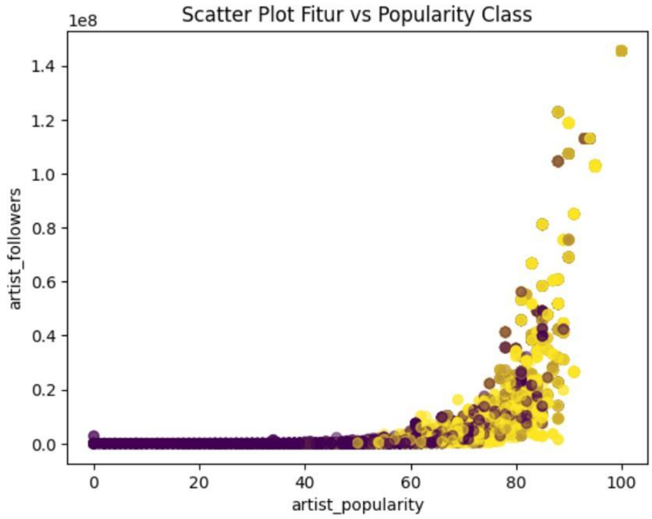


Figure 4 shows that there is a pretty clear grouping. The dots that show popular songs (usually light or yellow) tend to be in the upper right corner. This means that the artist has a lot of fans and is very well-known. In the lower left area, on the other hand, there are songs that aren't popular (dark/purple). This visual separation shows why the Random Forest algorithm uses these two features as important parts of its classification decisions.

CONCLUSION

Random Forest was the better classifier for predicting how popular a song would be on Spotify. It was 76.54% accurate, while Naive Bayes was only 72.21% accurate. This gap shows that the ensemble method works better with complex, nonlinear data patterns than Naive Bayes's strict independence assumptions. Because of this, ensemble methods are the best way to do similar data science tasks.

The analysis of feature importance revealed that an artist's established reputation, particularly artist_popularity and artist_followers, is the paramount factor in their success. In today's streaming world, "brand power" is much more important than structural metadata like album size or track count.

What to do next To improve these predictions, future research should incorporate intrinsic audio attributes such as tempo, key, and danceability into the dataset. Testing more complex architectures like Gradient Boosting or Neural Networks might also help find more subtle patterns and make the error rates even lower.

It enables a deeper understanding of which audio attributes contribute most significantly to genre classification. This insight can be useful for both model optimization and further music data analysis. For future work, this research can be expanded by evaluating additional machine learning or deep learning algorithms to obtain a broader performance comparison. Moreover, the application of feature selection or feature engineering techniques may help reduce data dimensionality and noise, potentially leading to improved classification accuracy and more efficient models.

REFERENCES

[1] **Visutsak et al. (2025)** Visutsak, P., Loungna, J., Sopromrat, S., Jantip, C., Soponkittikunchai, P., & Liu, X. (2025). Mood-Based Music Discovery: A System for Generating Personalized Thai Music Playlists Using Emotion Analysis. *Applied System Innovation, 8*(2), 37.

[2] **Lee & Tu (2025)** Lee, C.-Y., & Tu, Y.-N. (2025). Predicting Hit Songs Using Audio and Visual Features. *Engineering Proceedings, 89*(1), 43.

[3] **Shu (2024)** Shu, M. (2024). Exploring Spotify's Music Popularity Dynamics and Forecasting with

Machine Learning. Dalam *Proceedings of the 2nd International Conference on Applied Physics and Mathematical Modeling* (hlm. 83-91).

[4] **Yee & Raheem (2022)** Yee, Y. K., & Raheem, M. (2022). Predicting Music Popularity Using Spotify and YouTube Features. *Indian Journal of Science and Technology, 15*(36), 1786-1799.

[5] **Marsya et al. (2025)** Marsya, N. D., Munadhil, M. M., Dzakyananta, M. A., Amaliah, K., & Rofiantos, D. (2025). Penerapan Algoritma Random Forest dalam Prediksi Emosi Musik Berdasarkan Karakteristik Fitur Audio Spotify. *Jurnal Sains Informatika Terapan (JSIT), 4*(2), 435-440.

[6] **Bahadure et al. (2025)** Bahadure, N. B., Vishwakarma, M., Kurhekar, M. P., Patni, J. C., & Patil, P. D. (2025). Machine Learning-Based Music Classification and Recommendation System from Spotify. *International Journal of Computer Information Systems and Industrial Management Applications, 17*, 143-154.

[7] **Fakhriza et al. (2025)** Fakhriza, F., Subekti, D., & Cahyo, P. W. (2025). Optimalisasi Algoritma Random Forest - Feature Selection dan Hyperparameter Tuning Klasifikasi Genre Musik. *JIKA (Jurnal Informatika) Universitas Muhammadiyah Tangerang, 9*(1), 16-25.

[8] **Kamilah et al. (2024)** Kamilah, H., Oktaviana, E., Hariyanto, V., Robi'atul Adawiyah, N. A., & Imannulloh, D. T. (2024). Prediksi Tren Musik Tahun 2024 pada Platform Spotify Menggunakan Metode Random Forest Regressor. *Analitik dan Visualisasi Data*.

[9] **Matusevičius (2021)** Matusevičius, M. (2021). *User Churn Prediction within Music Streaming Service Industry: A Comparison of Machine Learning Models* (Tesis Master tidak diterbitkan). Tilburg University, Belanda.

[10] **Nasution (2024)** Nasution, M. Q. R. (2024). *Implementasi Algoritma Random Forest pada Sistem Rekomendasi Musik Menggunakan Teknik Collaborative Filtering* (Skripsi tidak diterbitkan). Universitas Medan Area, Medan.

[11] **Dhamara et al. (2025)** Dhamara, G. Z., Sucipto, & Nugroho, A. (2025). Klasifikasi Genre Musik Menggunakan Machine Learning. *Bulletin of Information Technology (BIT), 6*(3), 206-217 .

[12] **Fakhriza et al. (2025)** Fakhriza, F., Subekti, D., & Cahyo, P. W. (2025). Optimalisasi Algoritma Random Forest - Feature Selection dan Hyperparameter Tuning Klasifikasi Genre Musik. *JIKA (Jurnal Informatika) Universitas Muhammadiyah Tangerang, 9*(1), 16-25.

[13] **Fiqri et al. (2025)** Fiqri, M., Lizen, F. B. S., & Ikrom, M. (2025). Implementation of Supervised Learning Algorithm on Spotify Music Genre Classification. *IJATIS: Indonesian Journal of Applied Technology and Innovation Science, 2*(1), 7-12 .

[14] **Ginabila & Fauzi (2023)** Ginabila, & Fauzi, A. (2023). Analisis Sentimen Terhadap Pemutar Musik Online Spotify Dengan Algoritma Naive Bayes dan Support Vector Machine. *Jurnal Ilmiah ILKOMINFO - Jurnal Ilmu Komputer dan Informatika, 6*(2), 111-125.

[15] **Ginn & Johnson (2024)** Ginn, B., & Johnson, N. (2024). *Evaluating the Efficacy of Random Forest in Mood Classification for Music Recommendation System*.

[16] **Marlia et al. (2024)** Marlia, S., Setiawan, K., & Juliane, C. (2024). Analisis Fitur Musik dan Tren Popularitas Lagu di Spotify menggunakan K-Means dan CRISP-DM. *SISTEMASI: Jurnal Sistem Informasi, 13*(2), 595-607.

[17] **Navisa et al. (2021)** Navisa, S., Hakim, L., & Nabilah, A. (2021). Komparasi Algoritma Klasifikasi Genre Musik pada Spotify Menggunakan CRISP-DM. *Jurnal Sistem Cerdas, 4*(2), 114-125.

[18] **Saragih (2023)** Saragih, H. S. (2023). Predicting song popularity based on Spotify's audio features: insights from the Indonesian streaming users. *Journal of Management Analytics, 10*(4), 693-709 .

[19] **Shulhiyana & Voutama (2025)** Shulhiyana, S., & Voutama, A. (2025). Penerapan Metode K-Means Clustering untuk Pengelompokkan Musik Berdasarkan Karakteristik Audio. *Jurnal Nasional Ilmu Komputer, 6*(2), 1-10.

[20] **Wau & Fatmawati (2025)** Wau, R. E. S., & Fatmawati. (2025). Klasifikasi Sentimen Ulasan Aplikasi Spotify Di Google Play Store Menggunakan Algoritma C4.5. *IJCIT (Indonesian Journal on Computer and Information Technology), 10*(1), 1-9.

[21] **Anggoro & Izzatillah (2022)** Anggoro, M. V., & Izzatillah, M. (2022). Sistem Rekomendasi Musik dengan Metode Collaborative Filtering Berbasis Android. *STRING (Satuan Tulisan Riset dan Inovasi Teknologi), 7*(1), 94-102.

[22] **Beesa et al. (2023)** Beesa, P., Naregavi, V., Imandar, J., & Thatte, S. (2023). Songs Popularity Analysis Using Spotify Data: An exploratory study. *Vidhyayana - An International Multidisciplinary Peer-Reviewed E-Journal, 8*(2), 17-25.

[23] **Jiang (2023)** Jiang, S. (2023). Predicting Music Popularity: A Machine Learning Approach Using Spotify Data. Dalam *Proceedings of the 2023 International Conference on Computer Science and Artificial Intelligence* (hlm. 1-7). Huazhong University of Science and Technology.

[24] **Setiawan et al. (2024)** Setiawan, Y., Hadiana, A. I., & Umbara, F. R. (2024). Customer Churn Prediction Using the Random Forest Algorithm. *JIKO (Jurnal Informatika dan Komputer), 7*(3), 209-216.

[25] **Tannady et al. (2024)** Tannady, H., Andry, J. F., Honni, & Lee, F. S. (2024). Analisis Big Data Spotify dengan Metode Data Mining. *JBASE - Journal of Business and Audit Information Systems, 7*(2), 52-59.