**International Journal of Computer and Information System (IJCIS)**
Peer Reviewed – International Journal
**Vol : Vol. 07, Issue 01, March 2026**
**e-ISSN : 2745-9659**
https://ijcis.net/index.php/ijcis/index

# Comparison of Classical and Neural Network-Based Models for Sentiment Analysis of Spotify Reviews

1st Asy Syifaur Roisah Rufaida
*Program Studi Sistem Informasi, Fakultas Komunikasi dan Informatika*
*Universitas Muhammadiyah Surakarta*
Surakarta, Indonesia
asyifaur@ums.ac.id

***Abstract*— Digital music streaming platforms such as Spotify have transformed how users access and consume music, generating large volumes of user reviews that reflect satisfaction and dissatisfaction with the service. Sentiment analysis of these reviews can provide valuable insights for developers and service providers. Although a number of studies have performed sentiment analysis on Spotify reviews, most of them are limited to comparing only classical algorithms. As a result, there is still a lack of evaluation that compares several classical algorithms and a simple neural-network-based model under the same feature representation and standardized text preprocessing on the same dataset. Therefore, this study aims to: (1) build sentiment classification models for Spotify reviews using classical machine learning algorithms, namely Naive Bayes, Support Vector Machine (SVM), and Logistic Regression; (2) develop and evaluate a Multi-Layer Perceptron (MLP) model using the same TF-IDF feature representation; and (3) compare the performance of all models. The experimental results show that Logistic Regression consistently achieved the best overall performance, with the highest accuracy (87.85%), precision (87.79%), and F1-score (87.65%), slightly outperforming Linear SVM and clearly surpassing Naive Bayes and MLP. Although the MLP model obtained a slightly higher recall than Naive Bayes, its overall performance remained lower than that of Logistic Regression and Linear SVM.***

*Keywords* : Sentiment Analysis, Machine Learning, Neural-Network-Based, Spotify

## I. INTRODUCTION

Digital transformation has revolutionized the music industry with the emergence of streaming services that offer instant access to millions of songs [1]. The shift from listening to music through traditional media to digital platforms has created opportunities for application developers to build music service products. It is predicted that the annual revenue growth of music streaming service providers reached 26.2 million dollars in 2022, an increase of 9.0% [2]. This growth is driven by the expansion of paid streaming services with a focus on devices that will further boost market growth [3]. In addition, subscription streaming revenue increased by 10.3% at the end of 2022, marking the highest revenue level of all time [4].

One of the largest music streaming platforms, Spotify, has dominated the global market with more than 500 million monthly active users [1]. Spotify is a popular music streaming service founded by Daniel Ek and Martin Lorentzon in 2006 in Stockholm, Sweden. The service was officially launched on October 7, 2008 [5]. Spotify allows users to access a large catalog of songs, albums, and podcasts from various music genres around the world via the internet.

With the emergence of new technologies, there will naturally be many opinions from various audiences, including toward the Spotify application. The opinions or reviews of a group of people on the Play Store or App Store can influence others and affect the popularity of an application. Therefore, user opinions or reviews are important as feedback for developers of online music player applications. The opinions or reviews of users who have used this application can also serve as indicators of the quality of the application itself [5]. In the reviews provided by users, there are many experiences, both good and bad, reflected as positive, negative, or neutral sentiments; moreover, these reviews can influence potential users [6].

With the number of reviews reaching tens of millions, companies face major challenges in understanding patterns of user satisfaction and dissatisfaction manually. Manual analysis of large-scale data is not only time-consuming but also prone to bias and inconsistency. Therefore, an artificial-intelligence-based approach is needed to systematically extract patterns from these review data [7]. Analysis of user reviews can be carried out using sentiment analysis. Sentiment analysis is a process of determining a person's sentiment or opinion expressed in textual form, which can be categorized as either positive or negative sentiment [8].

Several studies on sentiment analysis of Spotify reviews have been conducted previously, and most of them use algorithms such as Support Vector Machine, Naive Bayes, and other machine learning methods. For example, study [1] employed the Naive Bayes algorithm and achieved an accuracy of 86.5%. Other studies compare two or more algorithms. Research [5] compared Naive Bayes and SVM, where Naive Bayes performed better with an accuracy of 84.73%. Another study [9], compared Naive Bayes and Logistic Regression, and the results showed that Logistic Regression outperformed Naive Bayes with an accuracy of 79%. Research [10] compared Random Forest, Support Vector Machine (SVM), and Naive Bayes. The evaluation results indicated that SVM delivered the best performance with an accuracy of 85.10%.

Although a number of studies have performed sentiment analysis on Spotify reviews using algorithms such as Naive Bayes, Support Vector Machine, Logistic Regression, and Random Forest, most of them are limited to comparing only classical algorithms. As a result, there is still a lack of evaluation that compares several classical algorithms and a simple neural-network-based model under the same feature representation and standardized text preprocessing on the same dataset. Therefore, this study aims to: (1) build sentiment classification models for Spotify reviews using

classical machine learning algorithms, namely Naive Bayes, Support Vector Machine (SVM), and Logistic Regression; (2) develop and evaluate a Multi-Layer Perceptron (MLP) model using the same TF-IDF feature representation; and (3) compare the performance of all models based on accuracy, precision, recall, and F1-score in order to identify the most effective and practically applicable approach for classifying positive and negative Spotify user reviews.

## II. RESEARCH METHODS

In this study, each stage is designed sequentially so that the research process can run properly, starting from data collection, preprocessing, analysis, and finally model evaluation. The research stages are illustrated in Figure 1.

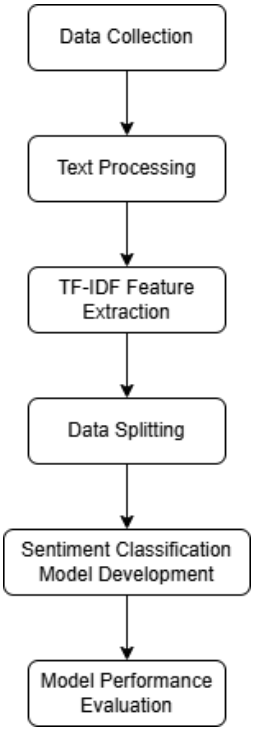


Figure 1. Research Stages

### 2.1 Data Collection

The dataset used in this study was obtained from Kaggle, a platform that provides various open resources commonly used by researchers. In this work, the author uses a database containing English-language Spotify user reviews. The dataset consists of a review text column and a sentiment label column with positive/negative classes. This research deliberately uses a labeled dataset to train all machine learning models employed. Example of some positive and negative reviews are presented in Table 1.

Table 1. Example of Some Positive and a Negative Reviews

| Review | Label |
|---|---|
| Great music service, the audio is high quality and the app is easy to use. Also very quick and friendly support | Positive |
| Please ignore previous negative rating. This app is super great. I give it five stars+ | Positive |
| I love the selection and the lyrics are provided with the song you're listening to! | Positive |
| It's a great app and the best mp3 music app I have ever used but there is one problem that, why can't we play some songs or find some songs? despite this the app is wonderful I recommend it. it's just the best. | Positive |
| An amazing music experience | Positive |
| This pop-up "Get the best Spotify experience on Android 12" is too annoying. Please let's get rid of this. | Negative |
| Really buggy and terrible to use as of recently | Negative |
| Dear Spotify why do I get songs that I didn't put on my playlist??? And why do we have shuffle play? | Negative |
| The player controls sometimes disappear for no reason. App restart forgets what I was playing but fixes the issue. | Negative |
| Still extremely slow when changing storage to external sd card.. I'm convinced this is done on purpose, spotify knows of this issue and has done NOTHING to solve it! Over time I have changed sd cards, each being faster in read, write speeds(all samsung brand). And please add "don't like song" so it will never appear again in my searches or playlists. | Negative |

### 2.2 Text Preprocessing

The next stage is text preprocessing, which is a series of steps to clean and prepare the text data so that it is suitable for analysis. This process includes:

*1) Case folding*

Converting all text to lowercase to normalize variations in writing, such as inconsistent use of uppercase and lowercase letters at the beginning, middle, or end of the text [9].

*2) Expanding contractions*

This step converts contracted forms into their full forms [9]. Contractions are shortened forms of two or more words by omitting one or more characters, for example “I’m” which comes from “I am”.

*3) Removal of digits, special characters, and punctuation*

This step removes digits, special characters, and punctuation such as !"#$%&'()*+,-./:;<=>?@[\]^_{|}~`, eliminating input that is not alphabetic or meaningful connectors, because such characters can make it harder for the computer to process the data [2].

*4) Tokenization*

Tokenization is the process of splitting sentences into separate words called tokens.

*5) Stopword removal*

Stopword removal is the process of eliminating very frequent words that do not carry significant meaning for sentiment extraction [5]. Examples include “and”, “a”, “an”, “in”, “of”, and similar words [5].

*6) Lemmatization*

This step is used in natural language processing to convert words into their base or dictionary form, for example the lemma of “better” is “good”[9].

### 2.3 TF-IDF Feature Extraction

Text that has been preprocessed is transformed into a numerical representation using Term Frequency–Inverse

Document Frequency (TF-IDF). TF-IDF consists of two components: TF, which counts how often a word appears in a document, and IDF, which measures how widely that word is distributed across the document collection. This study uses the sklearn.feature_extraction library in Python to compute TF-IDF. The TF-IDF weighting calculation implemented in sklearn is shown in Equation (1).

$$W_{t,d} = tf_{t,d} \times \ln\left(\frac{1+N}{1+df_t}\right) + 1 \qquad (1)$$

Keterangan:
$W_{t,d}$ : Term Frequency - Inverse Document Frequency
$tf_{t,d}$ : Jumlah kata (term) t pada dokumen d
$N$ : Jumlah dokumen teks
$df_{(t)}$ : Jumlah dokumen teks yang mengandung kata (term) t

### 2.4 Data Splitting

The dataset is then divided into 80% training data and 20% test data using a stratified split so that the distribution of positive and negative classes remains balanced in both subsets.

### 2.5 Sentiment Classification Model Development

Several supervised learning algorithms are trained using TF-IDF features as input, namely:
1) *Multinomial Naive Bayes,*
2) *Linear Support Vector Machine (SVM),*
3) *Logistic Regression,*
4) *Multi-Layer Perceptron (MLP Neural Network).*

Each model is trained on the same training data for a fair comparison.

### 2.6 Model Performance Evaluation

Each model is evaluated on the test data using the metrics accuracy, precision, recall, and F1-score. The evaluation results are presented in tabular form to facilitate comparison of performance across algorithms.

## III. RESULT AND ANALYSIS

### 3.1 Data Collection

The data for this study were obtained from Kaggle. The dataset consists of 52686 user reviews of the Spotify application from the Google Play Store for the period January–July 2022. These reviews were scraped to analyze user sentiment and feedback toward the application. Each review has been labeled as either "Positive" or "Negative" based on its sentiment. The sentiment distribution in the dataset is shown in Figure 2.

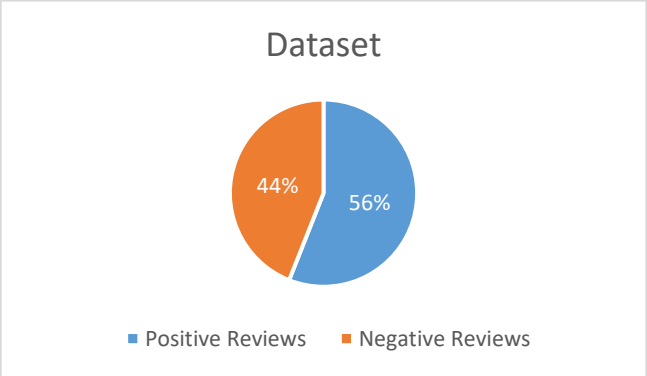


Figure 2. The Sentiment Distribution in the Dataset

Positive reviews: 56% of the total reviews
Negative reviews: 44% of the total reviews

### 3.2 Text Preprocessing

After collecting the data, the next step is data preprocessing, in which the collected data are cleaned and prepared for subsequent stages. This preprocessing stage consists of several steps:

1) *Case folding*

An example of the case folding results is shown in Table 2.

Table 2. Case Folding Result

| Before | After |
|---|---|
| **R**eally buggy and terrible to use as of recently | **r**eally buggy and terrible to use as of recently |

2) *Expanding contractions*

An example of expanding constractions results is shown in Table 3.

Table 3. Expanding Contractions Result

| Before | After |
|---|---|
| the consumer **doesn't** want to pay for music they **can't** fully own | the consumer **does not** want to pay for music they **cannot** fully own |

3) *Removal of digits, special characters, and punctuation*

An example of this step's results is shown in Table 4.

Table 4. Removal of Digits, Special Characters, and Punctuation Result

| Before | After |
|---|---|
| Dear Spotify why do I get songs that I didn't put on my playlist**???** | Dear Spotify why do I get songs that I didn't put on my playlist |

4) *Tokenization*

An example of tokenization results is shown in Table 5.

Table 5. Tokenization Result

| Before | After |
|---|---|
| Really buggy and terrible to use as of recently | 'Really', 'buggy', 'and', 'terrible', 'to', 'use', 'as', 'of', 'recently' |

5) *Stopword removal*

An example of stopword removal results is shown in Table 6.

Table 6. Stopword Removal Result

| Before | After |
|---|---|
| Really buggy **and** terrible **to** use **as of** recently | Really buggy terrible use recently |

*6) Lemmatization*

An example of lemmatization results is shown in Table 7.

Table 7. Lemmatization Result

| Before | After |
|---|---|
| Please **let's** get rid of this | Please **let** get rid of this |

The preprocessing results for the first ten data instances are presented in a table 8.

Table 8. Preprocessing Result

| Before | After |
|---|---|
| Great music service, the audio is high quality and the app is easy to use. Also very quick and friendly support. | great music service audio high quality app easy use also quick friendly support |
| Please ignore previous negative rating. This app is super great. I give it five stars+ | please ignore previous negative rating app super great give five star |
| This pop-up "Get the best Spotify experience on Android 12" is too annoying. Please let's get rid of this. | pop get best spotify experience android annoying please let u get rid |
| Really buggy and terrible to use as of recently | really buggy terrible use recently |
| Dear Spotify why do I get songs that I didn't put on my playlist??? And why do we have shuffle play? | dear spotify get song put playlist shuffle play |
| The player controls sometimes disappear for no reason. App restart forgets what I was playing but fixes the issue. | player control sometimes disappear reason app restart forgets playing fix issue |
| I love the selection and the lyrics are provided with the song you're listening to! | love selection lyric provided song listening |
| Still extremely slow when changing storage to external sd card.. I'm convinced this is done on purpose, spotify knows of this issue and has done NOTHING to solve it! Over time I have changed sd cards, each being faster in read, write speeds(all samsung brand). And please add "don't like song" so it will never appear again in my searches or playlists. | still extremely slow changing storage external sd card convinced done purpose spotify know issue done nothing solve time changed sd card faster read write speed samsung brand please add like song never appear search playlist |
| It's a great app and the best mp3 music app I have ever used but there is one problem that, why can't we play some songs or find some songs? despite this the app is wonderful I recommend it. it's just the best. | great app best mp music app ever used one problem cannot play song find song despite app wonderful recommend best |
| I'm deleting this app, for the following reasons: This app now has a failing business model. Whether streaming services like it, or not: the consumer doesn't want to pay for music they can't fully own, and 6 ads successively, upon logging in, before a single song, is too much. Closed the app during ad number 6, and I'm more patient than most. If those are the only ways you can profit: you've already peaked. All that's left is your decline. | deleting app following reason app failing business model whether streaming service like consumer want pay music cannot fully ad successively upon logging single song much closed app ad number patient way profit already peaked left decline |

### 3.3 Data Splitting

The preprocessed data were then split into 80% training data and 20% test data using a stratified split technique so that the distribution of positive and negative classes remains balanced in both subsets. The total number of instances used for training is 42148, while the number of test instances is 10538. The distribution of the data split is shown in table 9.

Table 9. The Distribution of the Data Split

| | Dataset | Train Data (80%) | Test Data (20%) |
|---|---|---|---|
| **Positive** | 23263 | 18610 | |
| **Negative** | 29423 | 23538 | 10538 |
| **Total** | 52686 | 42148 | |

### 3.4 Model Performance Evaluation

The evaluation stage shows the extent to which the constructed models can achieve satisfactory performance based on the evaluation metrics [11]. The evaluation results for each algorithm are presented in table 10 and illustrated in Figure 3.

Table 10. The Evaluation Results for Each Algorithm

| Algoritma | Precision | Recall | F1-Score | Accuracy |
|---|---|---|---|---|
| Naïve Bayes | 86.09% | 85.13% | 85.45% | 85.81% |
| Linear SVM | 87.20% | 87.07% | 87.13% | 87.33% |
| Logistic Regression | **87.79%** | **87.53%** | **87.65%** | **87.85%** |
| MLP Neural Network | 85.28% | 85.27% | 85.28% | 85.48% |

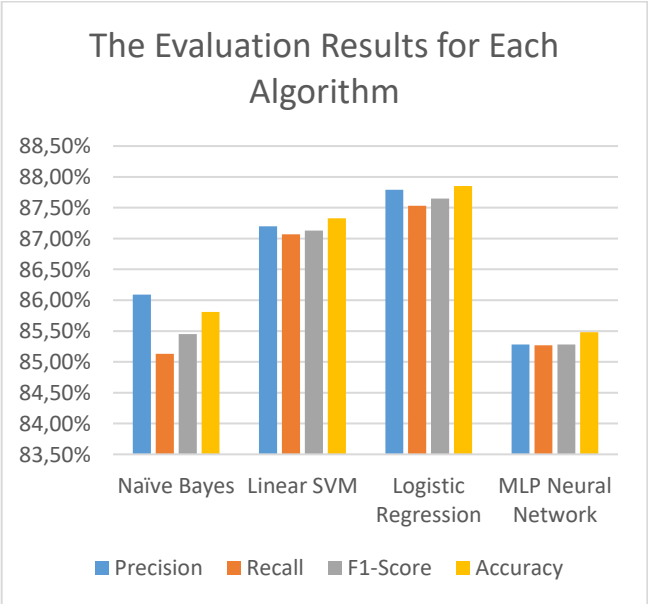


Figure 3. The Illustration of Evaluation Results for Each Algorithm

Based on table 10 and Figure 3, it can be seen that the algorithm with the highest accuracy is Logistic Regression, with an accuracy of 87.85%, followed by Linear SVM with 87.33%, Naive Bayes with 85.81%, and the lowest accuracy

obtained by the MLP Neural Network with 85.48%. The ranking of the algorithms is the same for the precision and F1-score metrics. The highest precision is achieved by Logistic Regression at 87.79%, followed by Linear SVM at 87.20%, Naive Bayes at 86.09%, and the lowest precision by the MLP Neural Network at 85.28%. Similarly, the highest F1-score is obtained by Logistic Regression at 87.65%, followed by Linear SVM at 87.13%, Naive Bayes at 85.45%, and the lowest F1-score by the MLP Neural Network at 85.28%. However, for the recall metric, the MLP Neural Network performs better than Naive Bayes. The order of recall values from highest to lowest is: Logistic Regression at 87.53%, Linear SVM at 87.07%, MLP Neural Network at 85.27%, and Naive Bayes at 85.13%. Based on these evaluation results, it can be concluded that the best algorithm for sentiment analysis of Spotify review data in this study is Logistic Regression.

The confusion matrices for each algorithm are presented in Figures 4–7. These matrices provide an overview of how well each model performs for the algorithms used.

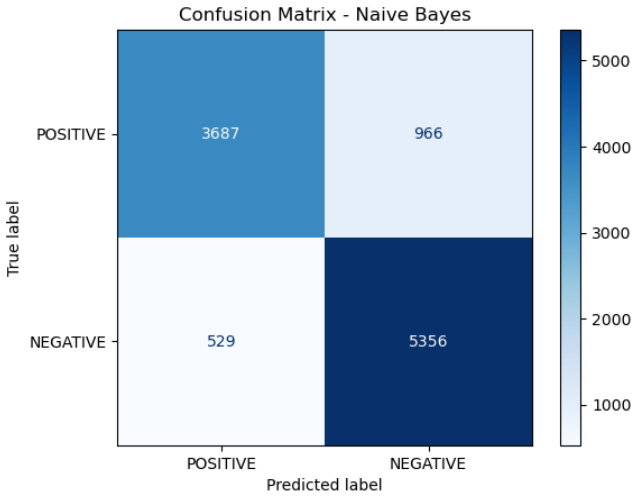


Figure 4. Confusion Matrix of Naïve Bayes

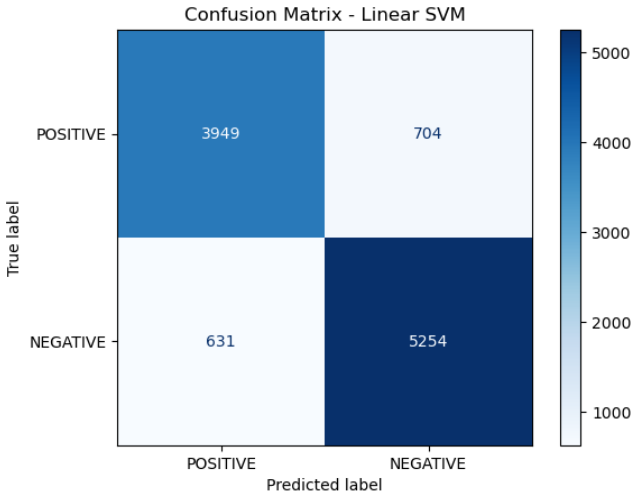


Figure 5. Confusion Matrix of Linear SVM

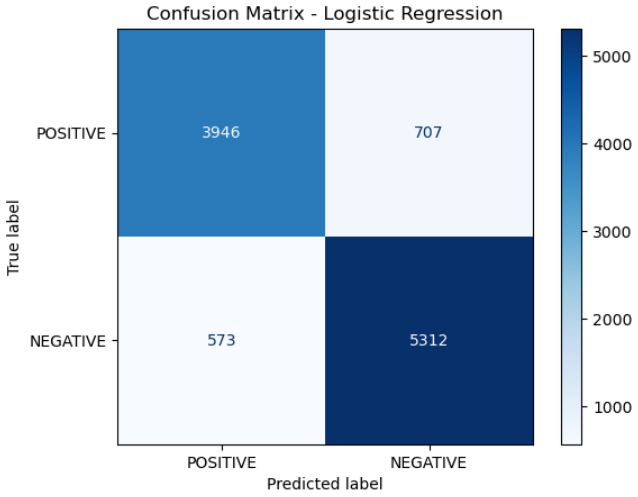


Figure 6. Confusion Matrix of Logistic Regression

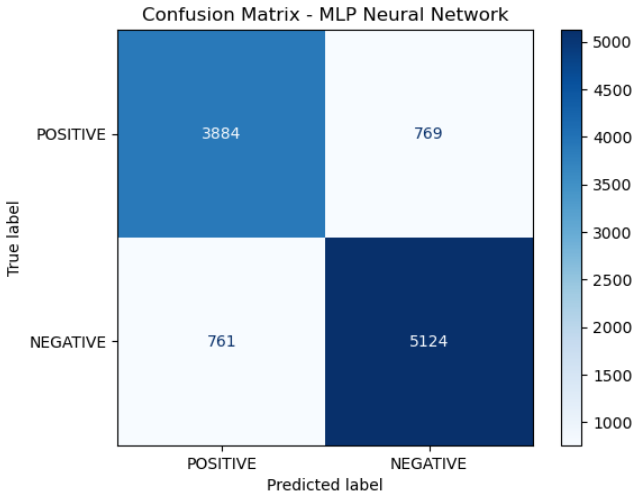


Figure 7. Confusion Matrix of MLP Neural Network

## VI. CONCLUSION

The experimental results show that Logistic Regression consistently achieved the best overall performance, with the highest accuracy (87.85%), precision (87.79%), and F1-score (87.65%), slightly outperforming Linear SVM and clearly surpassing Naive Bayes and MLP. Although the MLP model obtained a slightly higher recall than Naive Bayes, its overall performance remained lower than that of Logistic Regression and Linear SVM.

These findings indicate that, under a consistent TF-IDF representation and standardized preprocessing pipeline, Logistic Regression is the most effective and practically suitable algorithm for binary sentiment classification of Spotify user reviews in this setting. Future work may extend this study by exploring deep learning or transformer-based models, incorporating additional features, or applying the approach to other platforms and languages.

## REFERENCES

[1] W. R. As' ari, M. Arifin, D. L. Fithri, and P. Setiaji, "Analisis Sentimen Ulasan Pengguna Aplikasi Spotify Di Google Play Store Menggunakan Algoritma Naive Bayes," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 9, no. 2, pp. 3600–3607, 2025.

[2] B. H. Meilana, "Analisis Sentimen Ulasan pada Aplikasi Spotify Menggunakan Pendekatan Natural Language Processing".

[3] Roojai, "Gaya Hidup & Kesehatan. Sains membuktikan, ada banyak manfaat music untuk Kesehatan." [Online]. Available: http://www.roojai.co.id/article/gaya-hidup-kesehatan/

[4] businesswire, "Global Music Streaming Market 2021-2025: Market is Poised to Grow by $7.47 Billion, Progressing at a CAGR of 19% - ResearchAndMarkets.com." [Online]. Available: https://www.businesswire.com/news/home/20210323005840/en/Global-Music-Streaming-Market-2021-2025-Market-is-Poised-to-Grow-by-%247.47-Billion-Progressing-at-a-CAGR-of-19---ResearchAndMarkets.com

[5] Ginabila and A. Fauzi, "Analisis Sentimen Terhadap Pemutar Musik Online Spotify Dengan Algoritma Naive Bayes dan Support Vector Machine," *J. Ilm. Ilk.*, vol. 6, no. 2, pp. 111–122, 2023.

[6] E. Daryfayi and I. Asror, "Sentimen Analisis Pada Ulasan Google Play Store Menggunakan Metode NaÃ¯ ve Bayes," *eProceedings Eng.*, vol. 7, no. 2, 2020.

[7] S. Nafisyah and R. Sulistiyowati, "Analisis Sentimen Ulasan Produk Toko Online Esrocte untuk Peningkatan Pelayanan Menggunakan Algoritma Naïve Bayer," *Blantika Multidiscip. J.*, vol. 2, no. 8, 2024.

[8] N. M. S. Hadna, P. I. Santosa, and W. W. Winarno, "Studi literatur tentang perbandingan metode untuk proses analisis sentimen di Twitter," *Semin. Nas. Teknol. Inf. dan Komun*, vol. 2016, pp. 57–64, 2016.

[9] M. Prasetya, M. Wulandari, and S. A. Nikmah, "Implementasi NLP (Natural Language Processing) Dasar pada Analisis Sentiment Review Spotify," in *Seminar Nasional Teknologi & Sains*, 2024, pp. 145–153.

[10] A. L. Prastia and T. Asra, "Perbandingan algoritma random forest, svm, dan naïve bayes dalam analisis sentimen ulasan spotify di play store berbasis smote," *J. Inform. Kaputama*, vol. 10, no. 1, pp. 17–27, 2026.

[11] T. F. Rahmadanti, M. Jajuli, and I. Purnamasari, "Klasifikasi Pengguna Shopee Berdasarkan Promosi Menggunakan Naïve Bayes," *Gener. J.*, vol. 5, no. 2, pp. 81–90, 2021.