İSTANBUL UNIVERSITY PRESS

Acta Infologica,
ACIN 2025, 9 (1): 74–89

https://doi.org/10.26650/acin.1604272
Submitted: 19.12.2024
Revision Requested: 30.01.2025
Last Revision Received: 18.02.2025
Accepted: 24.02.2025
Published Online 26.02.2025

Acta Infologica

Research Article

Open Access

Optimizing the Sentiment Recognition in Spotify Playlists Through Ensemble-Based Approaches

M. Erdem İsenkul [1]

[1] İstanbul University-Cerrahpaşa, Faculty of Engineering, Department of Computer Engineering, İstanbul, Türkiye

Abstract Spotify, with over 320 million monthly active users reported in 2020, offers a unique platform for data science and machine learning applications. This study leverages Spotify's extensive music library of over 50 million songs to analyze the emotional tone of user-created playlists using machine learning algorithms. By employing advanced classification methods, including Random Forest, Decision Tree, and Support Vector Machines (SVM), the research compares their effectiveness in sentiment classification tasks. The Random Forest model achieved the highest test accuracy of 87%, closely followed by the Decision Tree model at 86%. These results highlight the potential of sentiment-informed data to enhance music recommendation systems by tailoring suggestions to users' emotional preferences. This work not only contributes to the evolving domain of sentiment-aware recommendation models but also demonstrates the technical challenges and practical implications of applying machine learning in music streaming platforms. The study's findings underscore the value of integrating emotional intelligence into recommendation algorithms to improve user engagement and satisfaction in digital music services.

Keywords Sentiment Analysis · Spotify API · Playlist · Machine Learning · Music Recommendation Systems

Citation: İsenkul, M. E. (2025). Optimizing the sentiment recognition in spotify playlists through ensemble-based approaches. *Acta Infologica*, *9*(1), 74-89. https://doi.org/10.26650/acin.1604272

Corresponding author: M. Erdem İsenkul eisenkul@iuc.edu.tr

Introduction

The proliferation of digital music platforms has transformed how individual access, curate, and experience music. Among these platforms, Spotify, with its expansive user base and extensive music catalog, stands out as a dynamic ecosystem for exploring user engagement, musical preferences, and emotional expression through playlist creation. As playlists are inherently personalized, they often reflect the moods and emotional states of their creators, offering an opportunity to delve into the sentiment embedded within them.

Sentiment analysis, a branch of natural language processing (NLP), traditionally used for understanding opinions in textual data, now finds meaningful applications in music streaming. Recognizing the emotional tone of playlists not only provides insights into user behavior but also enhances the quality of music recommendations by enabling streaming platforms to deliver more personalized content. Emotionally aware recommendations can improve user engagement, satisfaction, and overall streaming experience. Spotify, with its API offering vast data on playlists, songs, and user activity, provides an ideal environment to study sentiment classification in a musical context.

The primary aim of this study was to classify the sentiment of Spotify playlists using machine learning algorithms. By leveraging a dataset constructed from the Spotify API, we apply a range of classifiers, including DT, RF, and Support Vector Machines (SVM), to explore the sentiment within playlists and assess model performance. Through this research, we seek to contribute to the evolving field of sentiment-aware recommendation systems by addressing both technical challenges and practical applications in music streaming.

An important feature of this research is its exploration of the unique challenges of sentiment analysis on the playlist level of music streaming sites. The extraction of sentiment from the titles of songs is very much unlike sentiment analysis on well-structured text, where the titles of songs are generally brief, ornamental, and contextually reliant. Furthermore, the reasons behind people using playlists are multi-formed, making the overall sentiment determination of a playlist on its titles only complex. The use of sentiment classification by using diverse machine learning models is explored by this research, including its focus on sentiment classification on the playlist level, where very little is seen in the sentiment classification literature up to date. Contrasting previous sentiment classification literature that has mostly been on sentiment classification concerning tracks or lyrics, this contribution presents a new dimension by considering sentiment on the playlist level, hence bringing new ideas on sentiment-based recommendation of music. The findings of this research contribute to making emotionally aware recommendation systems better, hence enhancing the content personalization of streams of music.

This paper is structured as follows: Section 2 reviews the existing literature on sentiment analysis in music and user-generated content. Section 3 outlines the methodology, including data collection, pre-processing, feature extraction, and classifier implementation. Section 4 presents and discusses the experimental results, followed by Section 5, which explores the practical implications and potential for future research. The study concludes by summarizing the findings and their contributions to sentiment analysis and music recommendation systems.

Related Work

The intersection of music recommendation and sentiment analysis has spurred significant interest, leading to the development of various machine learning approaches that enhance user engagement by

personalizing the emotional tone of playlists. This section reviews the foundational work in sentiment analysis, its application in user-generated content such as playlists, and specific methodologies in music recommendation systems.

Sentiment analysis as a field has grown from the initial approaches in opinion mining and review classification. Hu and Liu's (2004) pioneering work established a framework for extracting sentiment from customer reviews, laying a foundation for text-based sentiment analysis in various domains. Pang and Lee (2008) further advanced this area with a comprehensive review of opinion mining methodologies, underscoring the evolution of sentiment classification techniques. Their work highlights the shift toward machine learning models and deep learning methods for handling the intricacies of sentiment within social media and user-generated content.

In the context of music, Chi et al. (2010) explored emotion-based playlist generation using reinforcement learning to create playlists that match specific emotional states. This early attempt to integrate sentiment into playlist curation demonstrated the potential for algorithmic approaches to enhance music personalization. Additionally, Jiang et al. (2011) examined sentiment classification on Twitter, expanding sentiment analysis to social platforms with user-generated data, further bridging the gap between NLP and user engagement applications. Abel et al. (2011) contributed to this domain by applying sentiment-aware models to Twitter for personalized news recommendations, presenting sentiment as a valuable feature in content recommendation systems.

Recent advancements have focused on incorporating sophisticated machine learning models into sentiment analysis for playlist recommendation. Delbouys et al. (2018) applied deep neural networks for mood detection in music based on audio and lyrical features, setting a benchmark for the integration of text and audio data in music mood analysis. Similarly, Schedl et al. (2021) provided a comprehensive overview of music recommendation systems, discussing the challenges and potential of emotion-aware recommendation models to capture the nuanced emotional preferences of users.

In addition to these foundational studies, Ferraro et al. (2018) explored hybrid recommendation systems that combine text and audio features, demonstrating the benefits of using multimodal data for playlist personalization. Their work, along with that of Irene et al. (2019), who used neural networks for automatic playlist generation, has informed modern approaches to sentiment-informed playlist creation. Both studies emphasize the role of feature extraction in accurately capturing the emotional context in music data.

To enrich user modeling, Pappas and Popescu-Belis (2016) developed sentiment-aware collaborative filtering, a method that personalizes recommendations based on user preferences for certain moods or sentiments. Their work, combined with the analysis of common-sense reasoning frameworks like ConceptNet by Liu and Singh (2004), supports the notion that incorporating external knowledge and user-centric features enhances recommendation accuracy in sentiment-driven applications.

Schedl's (2016) introduction of the LFM-1b dataset provided researchers with a valuable resource for evaluating sentiment analysis methods in large music datasets, contributing to benchmarking efforts in the field. The LFM-1b dataset continues to be widely used in music recommendation and sentiment analysis research, enabling the evaluation of models on real-world user behavior.

Within the current research landscape, our study stands out by applying an ensemble of machine learning models, including Random Forest, Decision Tree, and Support Vector Machines (SVM), to the sentiment classification of Spotify playlists. Leveraging Spotify's API and the Spotipy library, we constructed a dataset

of 2000 songs and applied both textual (VADER and TextBlob) and metadata-based feature extraction. By using machine learning classifiers with cross-validation techniques, including Leave-One-Subject-Out (LOSO) validation, our approach not only addresses playlist sentiment classification but also evaluates model robustness across user preferences, contributing to sentiment-aware recommendation systems in music streaming platforms.

In summary, this work builds upon existing sentiment analysis methodologies (Hu & Liu, 2004; Pang & Lee, 2008) and incorporates recent advances in music recommendation systems (Ferraro et al., 2018; Delbouys et al., 2018). By expanding sentiment analysis to the realm of playlist curation, we aim to enhance user engagement in music streaming applications, providing a foundation for further research in emotion-driven recommendation systems.

Material and Methods

This section details the methodology for extracting, preprocessing, and analyzing playlist data to classify sentiment in Spotify playlists. The approach integrates various natural language processing (NLP) techniques, feature extraction strategies, and machine learning models to ensure robust sentiment classification.

Dataset description

The dataset used in this study was curated by extracting songs from publicly accessible playlists on Spotify, leveraging the Spotify Web API to obtain a comprehensive set of features. The dataset comprises 2000 songs, carefully selected to include an equal number of tracks representing positive and negative emotions. This dataset forms the basis for exploring sentiment classification within Spotify playlists, with each song labeled according to its emotional tone, as supported by previous research on sentiment analysis of user-generated content (Abel *et al*., 2011; Chi *et al*., 2010).

To construct the dataset, the Spotipy library, a Python client for the Spotify Web API, was employed. Spotipy enables access to Spotify's extensive music catalog and metadata, including song titles, artist information, album details, and playlist characteristics. Through the Spotify API's RESTful architecture, JSON metadata about each song was retrieved, which included not only textual information but also user-defined playlist attributes, artist popularity, and genre distribution, enriching the dataset with contextual features (). Studies have shown that this type of metadata can enhance sentiment classification by providing a context that aligns with the emotional tone of user-generated playlists (Ferraro *et al*., 2018; Howells & Ertugan, 2017).

To achieve diversity along with the representativeness of the dataset, there is a mix of musical genres within the playlists. Particularly, there is around 30% of pop, 25% of rock, 15% of electronic, 10% of jazz, 10% of classical, and 10% of others, hence allowing diverse musical content within diverse categories. The diverse musical content is important to ensure that there is no bias that might result from domination by one kind of musical genre, while also promoting the overall generalizability of the sentiment classification schemes.

Additionally, the playlists were taken from users with diverse engagement levels ranging from occasional listeners to actively engaged editors of music constantly changing and adjusting respective playlists. The practice ensures that the dataset represents the diverse listening habits of people, hence making the findings applicable to real-world situations of recommending music.

Sentiment labeling

The labeling process was essential for categorizing songs as having either positive or negative sentiment. Two sentiment analysis libraries, VADER (Valence Aware Dictionary and sEntiment Reasoner) and TextBlob, were used to assess and label the emotional tone of each song title. VADER, specifically tailored for sentiment analysis in social media, was selected due to its ability to capture the nuances of informal language, making it well-suited for analyzing user-generated playlists on Spotify (Hu & Liu, 2004; Pang & Lee, 2008). TextBlob, another NLP tool, provided a secondary reference for sentiment labeling, offering valuable cross-validation insights. Ultimately, the VADER-labeled dataset was chosen for further analysis because of its higher alignment with user-perceived emotional tones in music.

Data preprocessing

Before applying the sentiment classification algorithms, the dataset underwent several preprocessing steps to optimize it for machine learning. This included standard NLP techniques such as tokenization, punctuation removal, and conversion of all text to lowercase. These steps were crucial for maintaining consistency across the dataset, enabling the machine learning models to accurately interpret and analyze song titles without unnecessary noise. To streamline this process, the NLTK (Natural Language Toolkit) library in Python was used, along with the string manipulation functions for cleaning symbols and extraneous characters. Such preprocessing is essential for sentiment classification, as demonstrated by Pang & Lee (2008) and Schedl, Knees, McFee & Bogdanov (2021), who highlight the importance of clean text inputs in improving classifier performance.

Feature extraction

Feature extraction focused on both textual and metadata attributes, ensuring a holistic approach to sentiment classification. Textual features were derived from song titles and artist names, using techniques such as bag-of-words and TF-IDF (Term Frequency-Inverse Document Frequency) to quantify word importance within the dataset (Turney, 2002). Additionally, word embeddings such as Word2Vec were used to capture semantic relationships, allowing the model to recognize and leverage similarities in sentimentally relevant vocabulary. Metadata features, including playlist creation dates, number of followers, and genre distribution, were incorporated to provide context that may influence playlist sentiment. Studies suggest that metadata features can add contextual value in sentiment classification, especially in user-generated music data (Delbouys *et al*., 2018; Liu & Singh, 2004).

Data distribution and balance

To ensure model reliability, the dataset was evenly divided, with 1000 songs labeled as positive and 1000 as negative. This balance was intentionally maintained to prevent bias toward either class, enhancing the model's ability to generalize across different emotional tones. The dataset spans various genres, including pop, rock, jazz, and electronic, reflecting several user preferences and increasing the robustness of the sentiment classification model (Schedl, 2016). Ensuring a balanced dataset aligns with best practices in machine learning, as imbalance can significantly affect classifier performance by introducing bias (Pappas & Popescu-Belis, 2016).

In summary, the curated dataset captures a diverse spectrum of musical content, enriched with both textual and contextual metadata. The inclusion of balanced sentiment labels, extensive preprocessing, and

feature extraction processes contribute to a comprehensive dataset that serves as a robust foundation for the sentiment classification of Spotify playlists.

Methodology

All experiments were conducted in Python using Jupyter Notebook as the primary development environment. The classification approach relied on implementing multiple machine learning models, each chosen for its strengths in handling text data and binary classification tasks, with a focus on extracting informative features and optimizing model performance through cross-validation and hyperparameter tuning.

The Random Forest classifier was selected for its robustness and ensemble nature, which improves accuracy by reducing overfitting. This model is particularly effective in handling complex datasets by combining multiple decision trees, yielding a more generalized model that performs well with high-dimensional data (Breiman, 2001; Delbouys *et al*., 2018). Random Forests have been shown to enhance performance in sentiment classification tasks by leveraging ensemble learning techniques, which combine predictions from multiple models to improve robustness and accuracy (Pang & Lee, 2008).

Support Vector Machines (SVM) were also applied, given their strength in high-dimensional spaces, which suits sentiment classification tasks where distinct boundaries between classes need to be defined. SVM's performance in text analysis tasks has been well-documented, with studies indicating its effectiveness in distinguishing between sentiment classes (Ferraro *et al*., 2018; Pappas & Popescu-Belis, 2016). SVMs are especially effective when the dataset involves clear class separations, as they seek to maximize the margin between classes, which has been beneficial in both text-based and multimedia sentiment analysis (Schedl, 2016).

Additionally, the Decision Tree classifier was used as a benchmark due to its interpretability and hierarchical structure, providing insights into feature importance and classification pathways (Pang & Lee, 2008; Turney, 2002). Decision Trees are often favored in sentiment analysis for their transparency, as they allow researchers to track classification decisions across different feature splits, contributing to explainable AI practices in recommendation systems (Jiang *et al*., 2011).

XGBoost, a gradient boosting algorithm, was included for its ability to enhance classification accuracy by iteratively correcting errors, which is advantageous for datasets with complex patterns. XGBoost's capacity to handle imbalanced data is well-documented, making it a valuable asset in sentiment analysis where certain emotions may be more prevalent (Chen & Guestrin, 2016; Schedl *et al*., 2021). XGBoost's iterative approach has proven to be highly effective in various domains of sentiment analysis, including music data, where its boosting mechanism addresses potential biases in training data (Howells & Ertugan, 2017).

Logistic Regression was applied as a straightforward linear model, valuable for its interpretability and the insights it provides regarding feature weights. Logistic Regression's scalability and fast training times of logistic regression make it a practical choice for initial classification testing in sentiment analysis, as its linear nature simplifies the interpretability of feature impact on class prediction (Abel *et al*., 2011; Hu & Liu, 2004).

Finally, Multinomial Naive Bayes was employed due to its computational efficiency and effectiveness with high-dimensional data, especially in text classification scenarios. Naive Bayes models have been widely used in sentiment classification, particularly when datasets include sparse text-based features, as they calculate class probabilities to handle high-dimensional spaces (Mumcuoğlu, 2007; Tarone *et al*., 2011).

This model's probabilistic approach serves as a reliable baseline for comparing the performance with more complex algorithms in sentiment analysis (Wang *et al*., 2020).

To ensure reliable evaluation, hyperparameter tuning was conducted for each model using grid search and cross-validation techniques. Leave-One-Subject-Out (LOSO) cross-validation was implemented to assess the generalizability of the models across diverse user preferences, thereby simulating real-world variability in playlist sentiment. This approach helped avoid model bias toward specific user data, a consideration that has proven crucial in similar sentiment classification research (Ferraro et al., 2018; Pang & Lee, 2008; Schedl, 2016). The comparative table for those methods is provided in Table 1.

Table 1

Comparative Analysis of the Classifiers

Classifier	Advantages	Disadvantages
Random Forest	• High accuracy and robustness • Handles large datasets with high dimensionality well • Reduces overfitting by combining multiple trees	• Computationally expensive and slow training • Lack of interpretability • Requires more memory
Logistic Regression	• Simple and easy to implement • Fast training and prediction • Provides probabilities for predictions	• Assumes a linear relationship between the features and output • Limited to linear decision boundaries • Sensitive to outliers
Decision Tree Classifier	• Intuitive and easy to understand • No need for feature scaling • Handles both numerical and categorical data well	• Prone to overfitting, especially on small datasets • Instability, small changes in data can result in a different tree • Lack of smoothness in the decision boundaries
Naive Bayes	• Simple and computationally efficient • Performs well with high-dimensional data • Fast training and prediction	• Assumes independence between features • Sensitivity to irrelevant features
Support Vector Machines (SVM)	• Effective in high-dimensional spaces • Versatile, different kernel functions for various data types • Effective in cases where the number of dimensions is greater than the number of samples	• Computationally expensive, especially with large datasets • Sensitivity to parameter tuning • Lack of transparency and interpretability

The training of these models involved a through exploration of hyperparameter tuning and optimization to enhance their performance. The ensemble of machine learning methods employed in this study aimed to provide a comprehensive understanding of sentiment dynamics within playlists, allowing for nuanced and accurate classification of emotional tones. This methodology lays the foundation for a detailed examination of sentiment patterns, contributing valuable insights to the intersection of machine learning and music sentiment analysis.

Classification

To evaluate the robustness and generalizability of our sentiment classification models, we employed the Leave-One-Subject-Out (LOSO) cross-validation strategy. In this approach, the dataset was divided into folds, each representing the playlists of a specific user. During each iteration, the data from one user was held out for testing, while the playlists from all other users were used to train the models.

This method is particularly valuable for playlist sentiment analysis because it closely resembles real-world conditions where sentiment patterns can vary significantly across individual user preferences. By systematically omitting one user's playlists in each iteration, the models were exposed to a range of user behaviors and musical tastes, enhancing their ability to learn generalizable sentiment dynamics within playlists. LOSO cross-validation enabled us to examine the models' capability to perform consistently across different users, ensuring that the sentiment classification was not overly influenced by specific user profiles. The aggregated results from each iteration offered a thorough assessment of the models' effectiveness in recognizing sentiment patterns across a diverse user base, thereby reinforcing the reliability and applicability of our sentiment analysis framework.

To comprehensively evaluate the performance of our sentiment classification models, we employed several key metrics: accuracy, precision, recall, F1 score, and Matthews Correlation Coefficient (MCC). These metrics provided insights into the models' ability to accurately classify sentiment across varying user playlists, while highlighting specific strengths and potential areas for improvement.

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{1}$$

where TP is the number of true positives, TN true negatives, FP false positives, and FN false negatives. Sensitivity and specificity are statistical measures that assess the accuracy of correctly classified positive and negative instances:

$$\text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{3}$$

The Matthews Correlation Coefficient (MCC) serves as a quality measure for binary classification in machine learning. It remains stable even when there are significant differences in class densities. MCC functions as a correlation coefficient between the predicted and observed binary classifications, yielding a value between 1 and +1. The formulation of the MCC metric is provided as follows:

$$\text{MCC} = \frac{(\text{TP} \times \text{TN}) - (\text{FP} \times \text{FN})}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \tag{4}$$

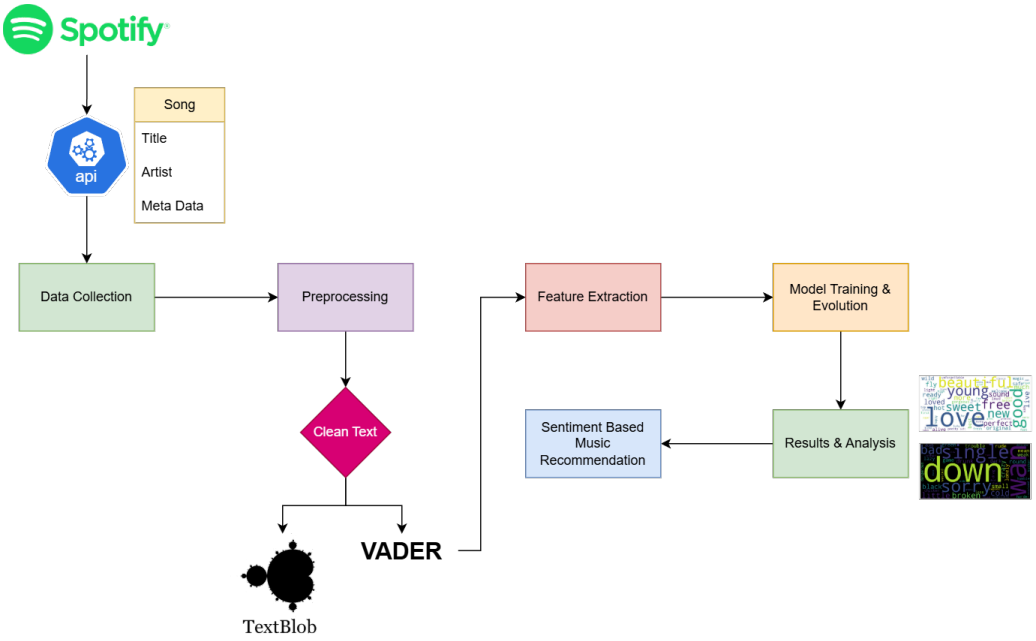
The Matthews Correlation Coefficient (MCC) attains a value of +1 when the classifier makes perfect predictions, –1 when the predictions and actual values are in complete disagreement, and 0 when the classification is no better than a random prediction. By using these metrics, we gained a holistic view of each model's performance, enabling us to identify the most reliable classifiers for sentiment classification in the playlist data. This multifaceted evaluation helped ensure that our models were both accurate and consistent in handling varied sentiment patterns across diverse user playlists.

To present an overall view of the proposed methodology, Figure 1 displays a flowchart that summarizes the key steps of the sentiment classification process. It incorporates acquisition of data through the Spotify

API, preprocessing measures like sanitizing of text and sentiment tagging, feature extraction processes using both textual features along with metadata features, and model training using diverse machine learning classifiers. The evaluation stage includes performance measures on diverse metrics like MCC to determine the most skilled classifier. Finally, the sentiment classification outcomes contribute to enhancing context-based music recommendations. The structured framework ensures that the resilience of the proposed approach is guaranteed along with its applicability.

Figure 1

Detailed Flowchart of the Proposed Method



Experimental Results

In this study, we compiled a dataset of 2000 songs and artist names using the Spotify API. To facilitate the dataset creation, a function was developed to identify and retrieve songs from user playlists by matching the playlist names. Specifically, we focused on a playlist titled *The Longest Playlist Ever*, created by user "Willis Orr" and set to public. This playlist was selected due to its extensive song collection, which provided a rich source of data. Additionally, another playlist containing primarily positive songs was created to serve as the test data.

Once the dataset was constructed, the data were labeled as positive or negative for sentiment classification. The sentiment labeling process employed two libraries, VADER and TextBlob, each of which provided sentiment polarity values for each song title. Before applying these libraries, the dataset underwent data cleaning procedures, including the removal of symbols and punctuation marks and the conversion of all text to lowercase using the NLTK library in Python.

After data cleaning, the sentiment polarity scores were calculated for both VADER and TextBlob, and the songs were labeled accordingly. The initial analysis indicated a predominance of positive sentiment within the data. However, VADER showed a higher frequency of negative labels compared to TextBlob, as shown in Figure 2 and Figure 3. Based on this comparison, we opted to proceed with the VADER-labeled dataset for further analysis.

Figure 2

Sentiment analysis results using VADER

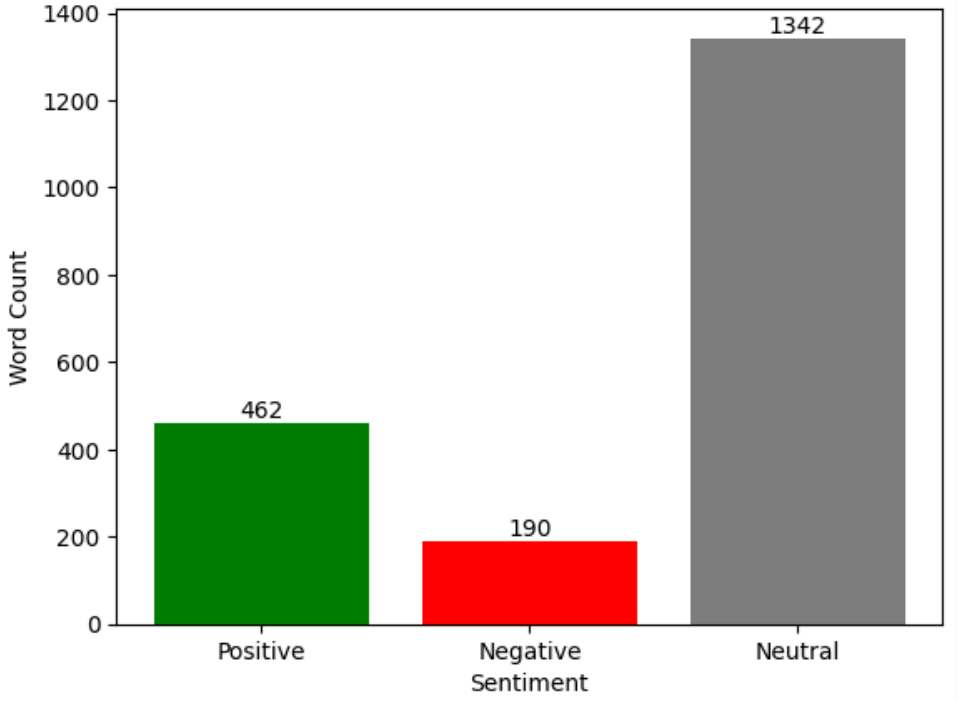
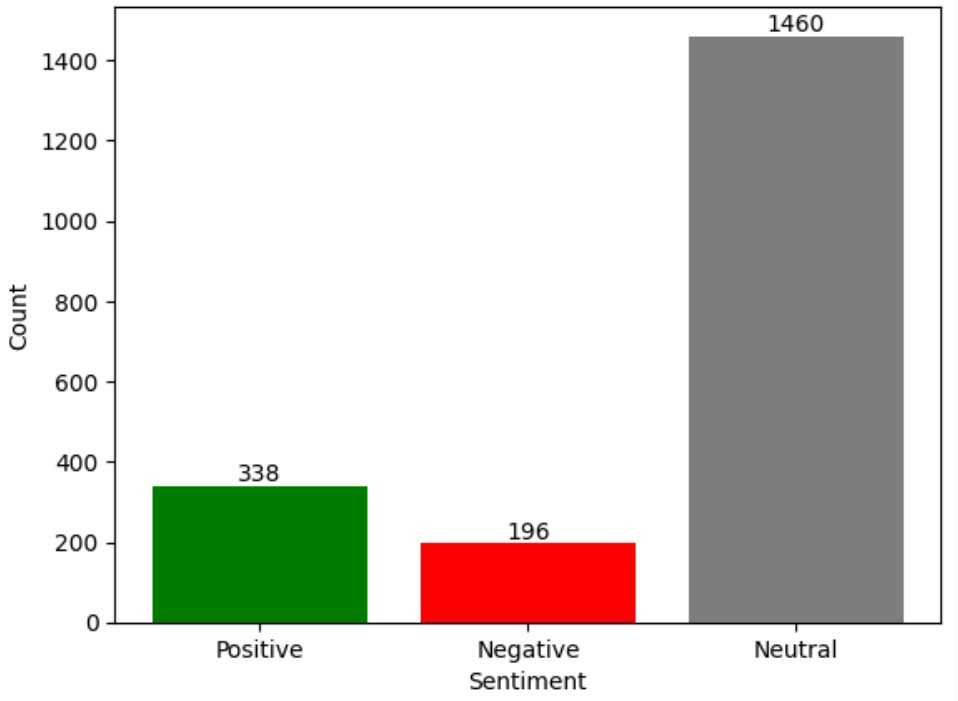


Figure 3

Sentiment analysis results using TextBlob



Following the labeling process, we calculated the frequency of the most commonly used words using count vectorization. The most frequently occurring word was "love," reflecting the common theme of happiness in music. Further analysis of word usage revealed distinct patterns in sentiment; for instance,

negatively labeled songs predominantly featured words such as "down," "sorry," "bad," "broken," and certain profanities (Figure 4 and Figure 5).

Figure 4

Word cloud of negative words

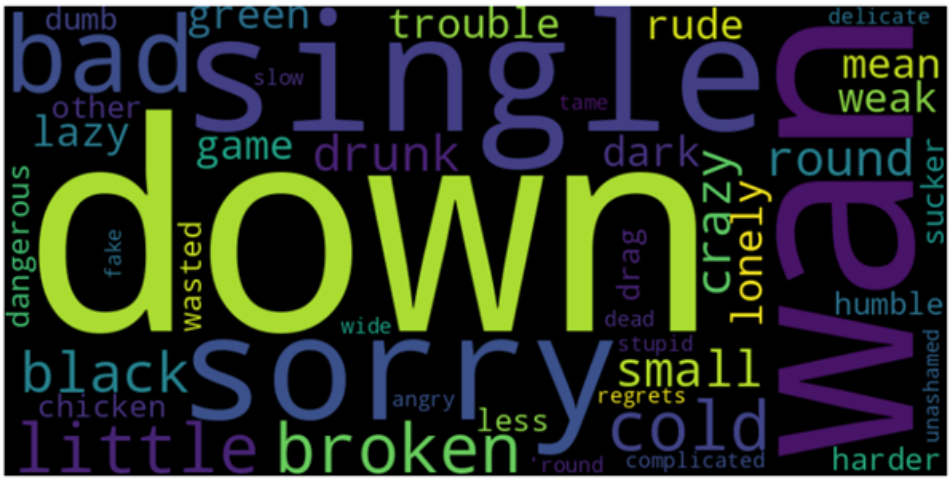
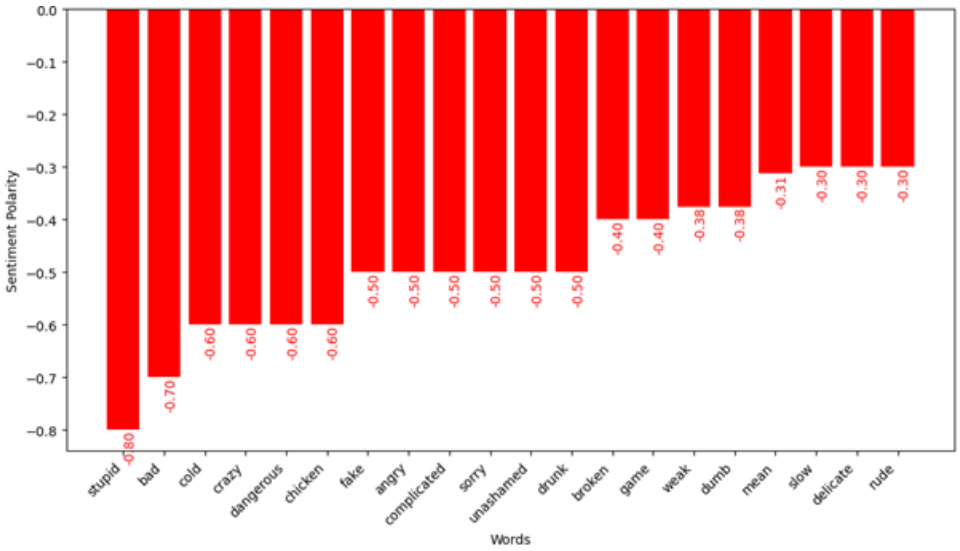


Figure 5

Top 20 words for highest negative sentiment polarities



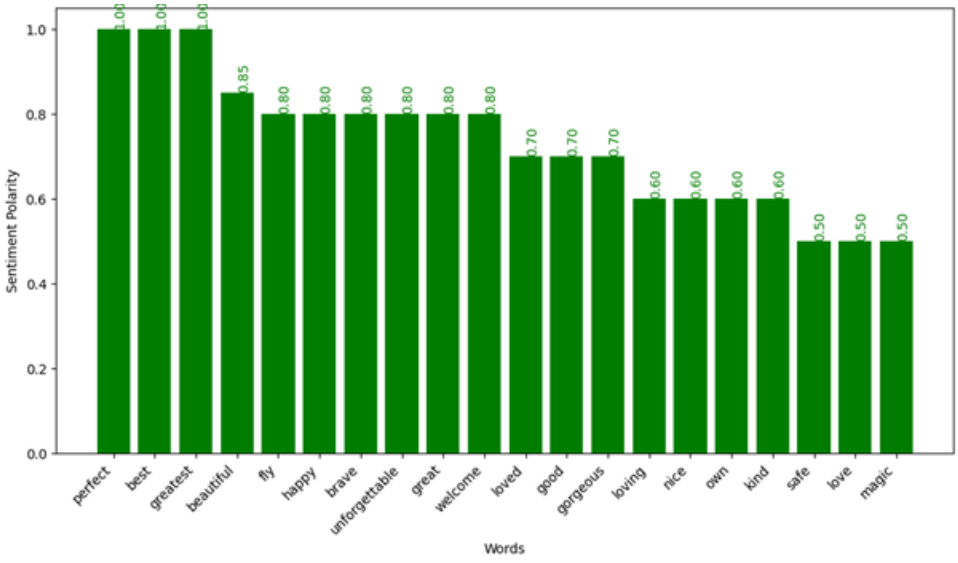
Conversely, positively labeled songs frequently contained words like "beautiful," "love", "sweet", "young", and "good" (Figure 6 and Figure 7). Based on these word frequency patterns, the dataset was transformed into a bag-of-words model, a representation commonly used in text classification for machine learning applications.

Figure 6

Word cloud of negative words

Figure 7

Top 20 words for highest negative sentiment polarities



For sentiment classification, six machine learning models were employed: Logistic Regression, Random Forest, Decision Tree, Support Vector Machine (SVM), XGBoost, and Multinomial Naive Bayes. Performance metrics, including accuracy, F1 score, recall, and precision, were calculated for each classifier to evaluate its effectiveness in sentiment classification (Table 2 and Table 3).

Table 2

Performance Metrics of Various Classifiers

Classifier	Training Accuracy	Validation Accuracy	Test Accuracy
Support Vector Machine	1.000	0.803	0.774
Logistic Regression	0.9311	0.757	0.741
Random Forest	1.000	0.840	0.866
Decision Tree	1.000	0.868	0.878

Classifier	Training Accuracy	Validation Accuracy	Test Accuracy
XGBClassifier	0.815	0.730	0.692
MultinomialNB	0.981	0.852	0.792

The training, validation, and test accuracies for each classifier are presented in Table 2. The SVM model achieved perfect training accuracy (1.000) but showed slightly lower validation (0.803) and test accuracies (0.774), suggesting potential overfitting. Logistic Regression attained a high training accuracy of 0.931, with validation and test accuracies of 0.757 and 0.741, respectively. The Random Forest classifier also achieved perfect training accuracy (1.000) and demonstrated strong validation (0.840) and test (0.866) accuracies. Similarly, the Decision Tree classifier achieved perfect training accuracy (1.000) and performed well on both the validation (0.868) and test (0.878) datasets. In contrast, the XGBClassifier showed a moderate training accuracy of 0.815, with lower validation (0.730) and test accuracies (0.692). Multinomial Naive Bayes reached a high training accuracy of 0.981, with validation and test accuracies of 0.852 and 0.792, respectively. Overall, the Decision Tree and Random Forest classifiers exhibited the highest test accuracies, indicating their robustness in generalizing to new data.

Table 3

Classifier Evaluation Metrics

Classifier	Precision	Recall	F1 Score	MCC
Support Vector Machine	0.711	1.000	0.826	0.577
Logistic Regression	0.683	1.000	0.807	0.509
Random Forest	0.833	0.991	0.903	0.722
Decision Tree	0.862	0.968	0.910	0.736
XGBClassifier	0.659	0.946	0.771	0.368
MultinomialNB	0.736	0.980	0.834	0.598

In terms of specific performance metrics, the Random Forest and Decision Tree classifiers demonstrated superior precision, recall, and F1 scores (Table 3). The Random Forest classifier achieved a precision of 0.833, recall of 0.991, and an F1 score of 0.903, resulting in a Matthews Correlation Coefficient (MCC) of 0.722. Similarly, the Decision Tree classifier displayed a precision of 0.862, recall of 0.968, and an F1 score of 0.910, with an MCC of 0.736. These results that the Random Forest and Decision Tree classifiers excel in achieving a balanced trade-off between precision and recall, making them particularly suitable for sentiment classification tasks in playlist data.

In addition to the F1-score, precision, recall, and accuracy, we have also added the Matthews Correlation Coefficient (MCC) as another metric of evaluation, considering its effectiveness to measure classification performance, especially where there is an imbalanced class situation. Unlike accuracy, where its result can have a lopsided view where one of the classes is dominant within the set of instances, MCC makes its result closer to fairness by considering false negatives, false positives, true negatives, and true positives. This characteristic makes it especially useful when there is a sentiment classification task under the imbalanced conditions of the class. With the use of MCC within our framework of evaluation, we make our classification performance evaluation richer and more reliable.

Discussion

The findings of this study highlight the varying effectiveness of machine learning models in classifying sentiment within Spotify playlists, with the random forest and decision tree classifiers emerging as the top-performing models. The overall success of these classifiers aligns with prior research in text-based sentiment classification, where ensemble methods such as Random Forest are often favored for their robustness in handling complex, multi-feature datasets (Breiman, 2001; Delbouys *et al*., 2018). The Random Forest model's ability to combine multiple decision trees allows it to capture subtle sentiment patterns that simpler models, such as Logistic Regression, may miss. The decision tree classifiers also demonstrated strong performance, achieving high test accuracy and an impressive balance between precision and recall. While individual Decision Trees are known to be prone to overfitting, the structured data preprocessing in this study likely minimized noise, allowing the model to generalize effectively across sentiment categories (Pang & Lee, 2008).

Support Vector Machines (SVM) also performed well, though with signs of overfitting, as evidenced by its high training accuracy and slightly lower validation and test accuracies. This phenomenon is common in SVM applications for text classification, where the high-dimensional nature of the data can challenge model generalizability (Pappas & Popescu-Belis, 2016, Sancar, 2024). SVM's reliance on well-defined margins works effectively for sentiment classes with clear separations; however, the ensemble nature of Random Forest allowed it to handle the varied sentiment expressions across playlists more flexibly. This observation aligns with that of Ferraro et al. (2018), who noted that ensemble approaches often outperform SVM in tasks involving diverse user data.

Logistic Regression achieved moderate accuracy, providing insights into feature weights yet potentially limiting its capacity to capture complex sentiment shifts within playlist data. Its linear nature made it less capable of handling nuanced patterns than non-linear models such as XGBoost and Random Forest. Studies on sentiment analysis have similarly shown that linear classifiers may struggle with data requiring deeper contextual interpretation (Hu & Liu, 2004).

XGBoost displayed mixed performance, with moderate training accuracy and lower validation and test accuracies compared to the other classifiers. While XGBoost is known for its strength in handling imbalanced datasets through iterative boosting, its performance here may indicate that the playlist sentiment data contained more complex variations than it could optimize for. Recent studies by Schedl et al. (2021) suggest that XGBoost performs well in mood-based music classification but may encounter limitations with polarity-focused sentiment analysis due to the subtleties involved in positive versus negative classification.

Multinomial Naive Bayes provided useful baseline insights but underperformed relative to other models. The Naive Bayes model assumes feature independence, an oversimplification that can limit its ability to capture sentiment nuances. This tendency to rely on individual word probabilities may explain why Naive Bayes struggles with sentiment classification in multi-dimensional data like playlists, as noted in prior sentiment analysis studies (Pang & Lee, 2008).

Recently, deep learning models have become increasingly prominent in sentiment analysis, particularly in multimedia contexts. Models like Recurrent Neural Networks (RNN) and their variants, such as Long Short-Term Memory (LSTM) networks, have shown notable success in capturing sentiment by considering sequential dependencies in text data (Zhang *et al*., 2022). In music sentiment analysis, Convolutional Neural Networks (CNNs) combined with audio features have further enhanced model performance, offering multi-

modal capabilities that integrate both text and audio for a richer sentiment profile (Birdal, 2024; Li *et al.*, 2023). While deep learning models require more computational power and extensive data for training, they hold promise for future applications in playlist sentiment classification, particularly when combined with ensemble approaches for increased accuracy and generalizability.

This study's use of the random forest and decision tree classifiers emphasizes the value of interpretable, high-performing models in sentiment classification. However, integrating deep learning models could advance this approach by capturing sequential patterns within playlists, enhancing both sentiment analysis and recommendation accuracy. Future work could explore hybrid models that blend traditional machine learning with deep learning, creating a multi-modal system that leverages both textual and acoustic features. Such an approach could significantly improve user engagement in music recommendation systems, addressing a key challenge in personalized content delivery.

In summary, while the random forest and decision tree models proved effective for playlist sentiment analysis, the potential of deep learning in this field remains considerable. This study contributes to the ongoing research in music recommendation by underscoring the value of sentiment-aware models, suggesting that future work may benefit from integrating ensemble and deep learning approaches to capture the complexities of user sentiment across diverse musical content.

Peer Review Externally peer-reviewed.
Conflict of Interest The author have no conflict of interest to declare.
Grant Support The author declared that this study has received no financial support.

Author Details M. Erdem İsenkul
[1] İstanbul University-Cerrahpaşa, Faculty of Engineering, Department of Computer Engineering, İstanbul, Türkiye
0000-0003-0856-2174 eisenkul@iuc.edu.tr

References

Abel, F., Gao, Q., Houben, G. J. & Tao, K. (2011). *Analyzing user modeling on Twitter for personalized news recommendations*. User Modeling, Adaption, and Personalization, 1–12. Springer.

Birdal, R. G. (2024). The Influence of Air Pollution Concentrations on Solar Irradiance Forecasting Using CNN-LSTM-mRMR Feature Extraction. *CMC-Computers Materials and Continua*, *78*(3), 4015-4028.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Chen, T. & Guestrin, C. (2016). *XGBoost: A scalable tree boosting system*. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794.

Delbouys, R., Hennequin, R., Piccoli, F., Royo-Letelier, J. & Moussallam, M. (2018). *Music mood detection based on audio and lyrics with deep neural net*. Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR), 688–695.

Ferraro, S., Bogdanov, D., Yoon, S., Kim, Y., & Serra, X. (2018). *Cross-cultural analysis of user behavior in music streaming services*. Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR), 1–6.

Hu, M. & Liu, B. (2004). *Mining and summarizing customer reviews*. Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 168–177.

Hutto, C. J., & Gilbert, E. (2014). *VADER: A parsimonious rule-based model for sentiment analysis of social media text*. Proceedings of the 8th International Conference on Weblogs and Social Media (ICWSM), 216–225.

Jiang, L., Yu, M., Zhou, M., Liu, X., & Zhao, T. (2011). *Target-dependent Twitter sentiment classification*. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 151–160.

Li, Y., Wang, X., Chen, J., & Xu, Z. (2023). Sentiment analysis on music recommendation: A convolutional neural network approach. *IEEE Transactions on Multimedia*, *25*(3), 689–700.

Liu, H., & Singh, P. (2004). ConceptNet: A practical commonsense reasoning toolkit. *BT Technology Journal*, *22*(4), 211–226.

Mumcuoğlu, K. Y. (2007). *Biotherapy laboratory protocol department of parasitology*. Israel: Hebrew University-Hadassah Medical School Jerusalem.

Pang, B. & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, *2*(1–2), 1–135.

Pappas, N. & Popescu-Belis, A. (2016). *Sentiment analysis of user comments for one-class collaborative filtering over text-based recommendations*. Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, 773–776.

Sancar, Y. (2024). Enhanced Classification of Skin Lesions Using Fine-Tuned MobileNet and DenseNet121 Models with Ensemble Learning. *Erzincan University Journal of Science and Technology*, *17*(3), 870-883.

Schedl, M. (2016). *The LFM-1b dataset for music recommendation and user modeling*. Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, 103–110.

Schedl, M., Knees, P., McFee, B., & Bogdanov, D. (2021). Music recommendation: State of the art and challenges. *IEEE Signal Processing Magazine*, *38*(3), 29–44.

Turney, P. D. (2002). *Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews*. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 417–424.

Zhang, J., Li, H., Zhao, R., & Huang, Y. (2022). Comparative study of LSTM and GRU for music sentiment analysis. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(5), 2345–2356.