DOI:10.14421/ijid.2025.4903 IJID (International Journal on Informatics for Development), *e-ISSN: 2549-7448*
Vol. 14, No. 1, June 2025, Pp. 530-546

Uncovering Insights in Spotify User Reviews with Optimized Support Vector Machine (SVM)

Nova Tri Romadloni*
Department of Informatics
Universitas Muhammadiyah Karanganyar
Karanganyar, Indonesia
novatrir@umuka.ac.id

Wakhid Kurniawan
Department of Informatics
Universitas Muhammadiyah Karanganyar
Karanganyar, Indonesia
kurniawan.wk48@gmail.com

Article History
Received December 25th, 2024
Revised February 2nd, 2025
Accepted February 3rd, 2025
Published April, 2025

***Abstract*— The rapid growth of user-generated reviews on platforms like Spotify necessitates efficient analytical techniques to extract valuable insights. This study employs a Support Vector Machine algorithm, optimized using Forward Selection, Backwards Elimination, Optimized Selection, Bagging, and AdaBoost, to effectively classify user reviews. A dataset of approximately 10,000 Spotify reviews was compiled from diverse online sources, ensuring a representative sample. The analysis reveals sentiment patterns across positive, negative, and neutral categories, with positive reviews dominates the landscape. These patterns help highlight Spotify's strengths while identifying areas for improvement. However, the SVM algorithm faces challenges in classifying minority classes, particularly negative sentiments, due to class imbalance. To address this, advanced optimization techniques are utilized to enhance classification precision and recall. Preprocessing steps, including data cleansing, tokenization, stemming, and stopword removal, refine the dataset, while TF-IDF converts text into numerical features for effective feature selection. The results show that the Optimized Selection method achieves the highest accuracy of 84.5%, outperforming other approaches. This research contributes significantly to developing balanced sentiment analysis models. Future studies may explore deep learning techniques to further improve classification accuracy and mitigate current limitations in data representation.**

Keywords*— *AdaBoost; backward elimination; bagging; forward selection; optimized selection

1 INTRODUCTION

In the digital age, music streaming platforms like Spotify have become an indispensable part of modern life. These platforms offer more than just music streaming services; they provide personalized experiences through features like curated playlists, tailored music recommendations, and social interactions that foster a sense of community among users. Within this ecosystem, user reviews are crucial in gauging satisfaction and preferences towards Spotify's services, providing developers with valuable feedback to drive service improvements and boost user engagement.

However, the sheer volume of user reviews on platforms like Spotify makes manual analysis impractical. To gain meaningful insights from this enormous pool of data, data-driven approaches, such as pattern analysis using machine learning algorithms, are essential. One of the most widely employed algorithms for sentiment analysis is the Support Vector Machine (SVM), a classification algorithm known for its ability to handle complex datasets with high accuracy.

In the context of user review analysis, the primary task often involves categorizing sentiments into different groups, such as positive, negative, and neutral. A significant challenge in this process is the issue of class imbalance, where negative sentiment reviews are typically underrepresented compared to other types. This imbalance can severely affect the accuracy and effectiveness of machine learning models. As a result, optimizing the algorithm to deal with these challenges becomes a crucial step, particularly when focusing on the minority class.

Various optimization techniques have been proposed to enhance SVM's performance. Techniques such as Forward Selection, Backwards Elimination, Optimized Selection, Bagging, and AdaBoost allow SVM to efficiently select the most relevant features, improve classification precision, and reduce computational complexity. These optimization methods, when integrated with SVM, can help the algorithm focus on the most impactful features of user reviews, such as specific terms indicative of positive or negative sentiment. The core innovation of this research lies in the integration of multiple SVM optimization techniques. While each optimization technique has its benefits, their combined application has the potential to yield superior results. This study focuses on utilizing a combination of Forward Selection, Backwards Elimination, Optimized Selection, Bagging, and AdaBoost to improve the accuracy and robustness of the SVM in sentiment analysis.

Forward Selection and Backwards Elimination are dimensionality reduction techniques that identify and retain the most impactful features in a dataset. By applying these methods, the algorithm can effectively eliminate irrelevant or redundant features, resulting in improved computational efficiency and model performance. Optimized Selection, on the other hand, ensures that the model prioritizes the most relevant attributes from the user reviews, leading to better classification results.

Ensemble methods such as Bagging and AdaBoost are integrated to enhance SVM's performance by addressing issues related to class imbalance. Bagging, or Bootstrap Aggregating, works by generating multiple versions of the training dataset through resampling and then training separate models on each of them. This process helps to reduce variance and improve prediction accuracy. AdaBoost, which stands for Adaptive Boosting, improves the model by focusing more on misclassified instances, effectively boosting the performance of the algorithm for difficult-to-classify data, especially for the minority class.

By combining these techniques, the research aims to create a more robust analytical framework for sentiment analysis in Spotify user reviews. This combination allows the model to handle imbalanced datasets more effectively while retaining the ability to capture key patterns in the data. Several studies have explored the optimization of SVM using various methods, and their findings offer significant insights into how these approaches can be applied to sentiment analysis. For instance, [1] demonstrates that genetic algorithms can be used to optimize the parameters of SVM for analyzing service reviews, resulting in a notable improvement in model accuracy. The study shows that incorporating a genetic algorithm to tune the parameters of the SVM algorithm led to higher performance when analyzing Go-Jek service reviews, ultimately improving classification results [1].

Similarly, Particle Swarm Optimization (PSO) has been shown to enhance sentiment analysis accuracy in specific use cases. For example, PSO combined with feature selection has led to a 95% accuracy rate in the analysis of legal documents, an improvement of almost 3% over the standard SVM approach [2]. This suggests that PSO, when integrated with SVM, improves the model's ability to identify significant patterns in large datasets, particularly in sentiment classification tasks.

Furthermore, research on PSO and its application to sentiment analysis highlights its effectiveness in addressing the problem of parameter optimization in SVM. In a study by Trifebi Shina Sabrila et al., PSO was used to optimize feature selection and improve sentiment analysis accuracy for customer reviews, resulting in an 18.84% improvement over the traditional SVM method [2]. This improvement underscores the utility of PSO in extracting relevant features, particularly in large-scale data analysis, which is highly relevant for platforms like Spotify.

Another critical study explored the application of Ant Colony Optimization (ACO) in optimizing SVM parameters for gearbox fault diagnosis, leading to improved model accuracy when compared to standard SVM methods [3]. These findings suggest that optimization techniques like ACO can be effectively applied to SVM, enhancing its performance across a range of applications, including sentiment analysis in user reviews. This indicates the versatility of ACO-based approaches in improving SVM's ability to classify data in real-world, dynamic environments. In the context of user reviews, combining PSO with Information Gain feature selection has been shown to improve the classification accuracy of sentiment analysis models. This combination is especially useful to analyze the sentiment of online reviews, as it helps identify the most relevant features from a vast array of data, improving the precision of the sentiment classification. This approach is relevant for platforms like Spotify, where user feedback can be diverse and complex [4].

While these optimization techniques have proven successful in improving SVM's accuracy, it is essential to critically analyze their application in real-world scenarios. For instance, while PSO has shown promising results in improving sentiment analysis for legal documents and customer reviews, its application to more complex and dynamic datasets, such as social media or music streaming platform reviews, requires further investigation. The impact of optimization on minority sentiment patterns, such as negative reviews, is also an area that warrants closer examination [5].

The application of SVM and optimization techniques to sentiment analysis of Spotify reviews offers a valuable opportunity to address several real-world problems. One of the key issues in analyzing Spotify reviews is the imbalance in sentiment distribution. Negative reviews, which are often critical of the platform's features, such as music recommendation algorithms, subscription pricing, and sound quality, may be underrepresented compared to positive reviews. This imbalance can skew the results of sentiment analysis, making it harder for the model to detect recurring themes of dissatisfaction.

By applying SVM optimization techniques, this research aims to improve the classification of negative sentiment reviews, allowing Spotify to better understand user complaints and concerns. For example, users may express dissatisfaction with the algorithmic recommendations, which they feel are not personalized enough. By accurately classifying and analyzing these negative reviews, Spotify can identify potential areas for improvement in their recommendation engine.

In addition to class imbalance, another challenge in analyzing Spotify reviews is the sheer diversity of user feedback. With millions of active users, Spotify receives a wide range of opinions, each reflecting unique preferences and concerns. These reviews can cover a broad spectrum of topics, including the music library, audio quality, app usability, subscription pricing, and more. By applying feature selection and optimization techniques, this study aims to extract the most relevant features from the data, focusing on the aspects that have the most significant impact on user experience.

Moreover, analyzing Spotify user reviews can uncover patterns related to user preferences, which can be valuable for Spotify's business strategy. For instance, user dissatisfaction with the pricing of subscription plans could indicate a need for more flexible pricing options. Similarly, complaints about algorithmic biases in recommendations may suggest that the platform needs to refine its recommendation algorithms to better match individual tastes.

The findings of this study can help Spotify not only improve its service but also enhance its overall user experience. By addressing the recurring themes in user feedback and focusing on the areas that matter most to users, Spotify can strengthen its position in the highly competitive music streaming market.

Recent research has explored additional optimization techniques to enhance SVM's performance further. A study by [6] showed that combining lexicons from SentiWordNet with SVM resulted in superior performance compared to using SVM alone. This hybrid approach helps in capturing the sentiment behind words more effectively, improving the accuracy of sentiment classification, particularly in reviews from online marketplaces. Similarly, [7] demonstrated that integrating Active Learning with SVM could optimize sentiment analysis, especially for big data applications in the tourism sector. Active Learning allows the model to iteratively select the most informative samples, improving performance by focusing on difficult-to-classify instances. Moreover, studies like those of [8] and [9] have explored the application of genetic algorithms to optimize SVM's accuracy in different domains. [8] used a genetic algorithm to analyze Apple product reviews, leading to a 15.76% accuracy improvement over the standard SVM method. In a different context, [9] applied PSO-based optimization for short-term electricity load forecasting, highlighting how optimized SVM can be used for dynamic data analysis. These studies indicate that hybrid methods combining optimization techniques can lead to more robust and reliable sentiment analysis models.

This research contributes to the growing field of sentiment analysis by leveraging advanced optimization techniques to improve SVM's performance in analyzing user reviews. By combining techniques such as Forward Selection, Backward Elimination, Optimized Selection, Bagging, and AdaBoost, this study aims to develop a robust model for accurately classifying sentiments in Spotify user reviews. The integration of these techniques addresses challenges such as class imbalance.

2 METHOD

The method section describes the step-by-step process undertaken in the research, illustrated through a clear procedural flow. Fig. 1 shows the method.

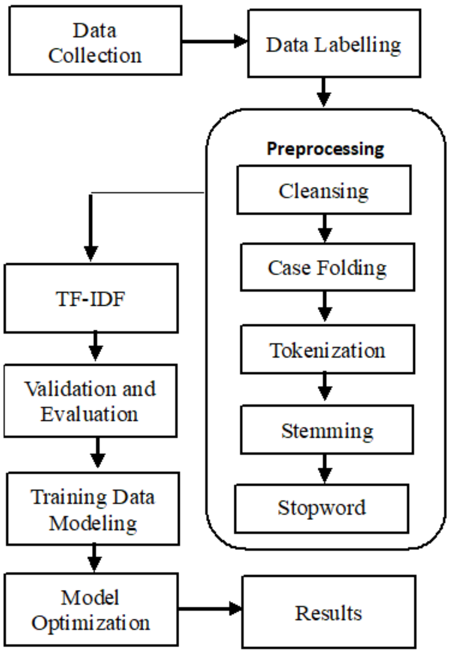

Figure 1. Research stages

Figure 1 provides a visual representation of the research design, illustrating the sequential steps from data collection, preprocessing, feature extraction, model training, and evaluation. Each stage is interconnected to achieve the primary objective of improving sentiment classification accuracy for minority classes, ensuring that the model effectively captures sentiment patterns in Spotify user reviews. By structuring the process systematically, this approach delivers deeper insights into user sentiment, which is crucial for developing more responsive data-driven services. Additionally, the model optimization techniques, including Forward Selection, Backwards Elimination, and Bagging, enhance classification performance, addressing class imbalance and facilitating research replication.

2.1 Data Collection

Spotify user reviews were collected from online sources such as discussion forums, social media platforms, and public review applications, with the primary aim of gathering a comprehensive dataset that reflects diverse user experiences and opinions. By leveraging multiple platforms, the data collection ensured a broad spectrum of user feedback, capturing sentiments from various demographics, including casual listeners and dedicated subscribers.

The data collection was conducted in 2024, from January to November, utilizing the Spotify Scraper API accessed via Google Colab. The API URL used for scraping reviews is https://api.spotify-scraper.com/v1/reviews, allowing automated and scalable retrieval of publicly available Spotify review data. Through this endpoint, review data such as review text, ratings, and metadata was extracted in a structured format. Keywords such as 'Spotify review,' 'playlist recommendations,' and 'audio quality' were strategically selected based on their relevance to Spotify's key service features and common user feedback topics. These keywords ensured that the collected data captured meaningful insights related to user experiences. Advanced filtering mechanisms were implemented to exclude irrelevant data, such as duplicate entries or reviews containing unrelated topics. The resulting dataset consisted of approximately 10,000 reviews, structured into columns containing review text (content), ratings, timestamps, and metadata, which served as the foundation for sentiment analysis. The sampling approach ensured a diverse representation of user opinions, including reviews from both free and premium users. A stratified sampling method was employed to maintain proportional representation across sentiment categories, ensuring that insights drawn from the analysis are reflective of broader user trends.

To address the issue of class imbalance, techniques for data augmentation were applied to improve the representation of minority classes (e.g., negative and neutral sentiments). The Synthetic Minority Oversampling Technique (SMOTE) was used to generate synthetic examples by interpolating between existing minority class samples, effectively increasing their representation without adding noise. Additionally, Random Over-Sampling was employed to duplicate minority class examples, while Text Augmentation techniques, such as synonym replacement and word swapping, were used to diversify the textual content of these classes. These approaches collectively enhanced the model's ability to classify minority classes accurately and balanced the dataset for training.

This study contributes to the development of more accurate and efficient sentiment analysis models by integrating multiple optimization techniques to enhance SVM performance. The implementation of SMOTE and text augmentation not only balances the dataset but also improves classification accuracy, particularly for underrepresented sentiment classes. These improvements provide a scalable approach that can be applied to other platforms dealing with user-generated content, offering a more balanced and insightful sentiment classification framework.

To ensure data quality and authenticity, additional preprocessing steps were conducted. Reviews from verified accounts and sources were prioritized to minimize the inclusion of spam or bot-generated content. Metadata, including timestamps and geographical details, was captured to provide additional context for analysis. The comprehensive dataset allowed for a detailed examination of sentiment patterns across regions and periods, aligning to capture user feedback on Spotify's latest features, pricing, and services during the specified period.

2.2 Data Labelling

The labeling process was essential to categorize the collected Spotify user reviews into sentiment classes: positive, neutral, and negative. The sentiment labels were determined based on the user ratings provided in the reviews. This method ensured a consistent and structured approach to defining sentiment categories, reducing ambiguity and enhancing the reliability of the analysis. Each review's sentiment label was carefully assigned to reflect the user's overall impression of Spotify's services.

Reviews with ratings of 1 and 2 were labeled as negative, indicating dissatisfaction or criticism. These reviews often highlighted specific pain points, such as technical glitches, dissatisfaction with pricing models, or limited song availability. On the other hand, a rating of 3 was classified as neutral, representing a balanced or indifferent perspective where users neither expressed strong approval nor discontent. Meanwhile, ratings of 4 and 5 were categorized as positive, reflecting satisfaction or praise for Spotify's services. Positive reviews typically included commendations on features like personalized playlists, seamless user interfaces, or high-quality audio streaming.

To enhance the labeling process, a manual review was conducted on a subset of the data to validate the automated categorization. This step ensured that sentiment labels aligned with the contextual meaning of the reviews, especially for borderline cases where ratings alone might not fully capture the sentiment. For example, a user might rate Spotify highly but mention drawbacks in their comments, which require additional scrutiny. This hybrid approach, combining automation with manual validation, reduced the risk of mislabeling and improved the quality of the labeled dataset.

The labeled dataset served as the foundation for the sentiment analysis model. It allowed the SVM algorithm to learn patterns from the labeled data and classify new reviews accurately. The structured dataset provided balanced input, enabling the model to distinguish subtle differences in sentiment expressions. Furthermore, this labeling strategy ensured the analysis captured the nuances of user sentiment effectively, aiding in actionable insights for improving Spotify's services. These insights were crucial for identifying areas requiring immediate attention, such as addressing recurring complaints or enhancing features highly valued by users.

2.3 Preprocessing

Preprocessing is a critical step in preparing text data for analysis. This phase involves transforming raw, unstructured data into a clean and structured format suitable for machine learning algorithms. Through preprocessing, irrelevant or redundant information is removed, thereby improving the efficiency and accuracy of subsequent analysis. It forms the backbone of text-based machine learning tasks, enabling models to focus on meaningful patterns and relationships within the data [1].

In the context of this research, preprocessing includes multiple stages such as cleansing, case folding, tokenization, stemming, and stopword removal. Each of these steps plays a unique role in refining the text data. For example, cleansing removes unwanted symbols or characters, such as punctuation marks, emojis, and hyperlinks, which often add noise to the dataset. Case folding standardizes text by converting it to lowercase, eliminating inconsistencies caused by capitalization. Stemming reduces words to their root forms, ensuring that variations of a word, such as "running" and "ran," are treated as the same term. This step is particularly crucial in sentiment analysis, where linguistic variations can skew the model's understanding.

An additional step in preprocessing is lemmatization, which is sometimes employed alongside or instead of stemming. While stemming truncates words to their root form, lemmatization considers the grammatical context to produce meaningful base forms. For instance, lemmatization transforms "better" into "good" based on its comparative context, providing more semantic clarity. This approach is especially beneficial for sentiment analysis, where nuanced language plays a significant role in expressing opinions. By integrating lemmatization, the research ensures that the textual data is not only uniform but also contextually accurate [2].

By systematically preprocessing the data, the model becomes less prone to errors caused by inconsistencies in the input data. Additionally, this step reduces the dimensionality of the dataset, making it computationally efficient and improving the performance of algorithms like Support Vector Machine (SVM). The streamlined dataset also enhances the interpretability of the results, allowing the research to identify key sentiment trends with greater precision. Moreover, preprocessing lays the groundwork for feature extraction techniques such as TF-IDF, ensuring that the text data is optimally prepared for analysis and classification.

2.3.1 *Cleansing and Case Folding:* Cleansing involves removing unwanted elements from the text, such as HTML tags, special characters, emojis, numbers, and URLs. These elements do not contribute to the sentiment of a review and can create noise in the analysis. For instance, links and hashtags in a review are often irrelevant to the actual sentiment being expressed and are removed during cleansing. Removing these elements ensures that the dataset remains focused on meaningful text, improving the clarity and reliability of the sentiment analysis process. Automated scripts and tools, such as regular expressions in Python, are often employed for this task to ensure consistency and efficiency in large datasets [3].

In addition to basic cleansing, advanced methods can be employed to handle more complex forms of noise. For example, spell correction algorithms may be used to standardize misspelled words, ensuring that variations do not fragment the dataset. Similarly, domain-specific stopwords, such as repetitive promotional phrases like "limited offer" or "premium subscription," can be removed to prevent them from skewing the analysis. By incorporating these advanced techniques, cleansing goes beyond removing visible noise and addresses subtler issues that could impact the quality of insights.

Case folding converts all text to lowercase, ensuring consistency in data representation. For example, "Spotify," "spotify," and "SPOTIFY" are treated as identical terms after case folding. This step simplifies text comparison and avoids redundant features caused by variations in capitalization. Furthermore, case folding helps standardize the input data, which is particularly important for natural language processing tasks that rely on frequency-based features, such as term frequency-inverse document frequency (TF-IDF) or word embeddings. By ensuring uniformity, case folding reduces computational overhead and enhances the accuracy of the analysis.

Cleansing and case folding together help streamline the dataset by removing inconsistencies and irrelevant components. This lays the foundation for subsequent preprocessing steps like tokenization and stemming, which operate on a cleaner and more uniform dataset. The combination of these techniques ensures that the text data is both readable and meaningful, providing a robust base for feature extraction and machine learning algorithms. Moreover, the streamlined dataset minimizes the risk of introducing errors or biases during analysis, enabling models like Support Vector Machine (SVM) to focus on the key aspects of sentiment conveyed in the reviews.

2.3.2 *Tokenization:* Tokenization splits a text into smaller units, typically words or phrases, known as tokens. For example, the sentence "Spotify offers great music recommendations" is tokenized into individual words: ["Spotify," "offers," "great," "music," "recommendations"]. Tokenization allows the algorithm to focus on the meaning of individual terms within the text, enabling the analysis of both the overall sentiment and specific topics addressed in the review [4]. This step is a foundational aspect of natural language processing (NLP), serving as the bridge between raw textual data and computational analysis [5].

There are different approaches to tokenization, such as word-based, character-based, and subword-based tokenization. Word-based tokenization, as used in this research, breaks text into individual words and is suitable for tasks like sentiment analysis and document classification. Character-based tokenization, which divides text into individual characters, is often used for languages without clear word boundaries or in cases involving misspellings. Subword-based tokenization, employed by advanced language models like BERT, splits text into smaller units to capture the nuances of complex word forms, making it ideal for multi-language applications [6].

Tokenization not only prepares text for analysis but also optimizes downstream processes like stemming, lemmatization, and stopword removal. By breaking text into tokens, the process reduces the dimensionality of the dataset, improving computational efficiency and model performance. Libraries such as NLTK and spaCy, widely used in this study, provide robust tools for tokenization, including customizable tokenization rules that adapt to specific data requirements [7]. This flexibility ensures that tokenization supports the unique challenges of analyzing diverse datasets, such as reviews from multilingual user bases.

2.3.3 *Stemming:* Stemming reduces words to their root forms, which helps in unifying variations of a word into a single representation. For instance, words like "playing," "played," and "plays" are stemmed to their root word, "play." This reduces redundancy and focuses the analysis on the core meaning of the terms. By simplifying the representation of words, stemming ensures that the dataset remains compact and meaningful, particularly in large-scale text analysis tasks [8].

Different stemming algorithms are designed to address various linguistic needs. The PorterStemmer, for example, is one of the most commonly used algorithms in English language processing. It operates based on a set of rules that iteratively strip suffixes from words, ensuring an accurate reduction to their stems. On the other hand, SnowballStemmer, a more advanced version of PorterStemmer, incorporates additional rules and support for multiple languages, making it a versatile choice for multilingual datasets [9]. These algorithms are essential for natural language processing tasks such as sentiment analysis, text classification, and information retrieval.

The benefits of stemming extend beyond dimensionality reduction. By consolidating words with similar meanings, stemming enhances the model's ability to generalize and reduces the risk of overfitting caused by treating morphological variants as distinct features [10]. This is particularly valuable in sentiment analysis, where similar words often convey identical sentiments. For example, "enjoyed" and "enjoying" would both contribute to a positive sentiment, and stemming ensures that they are treated equally. By aligning the dataset to its semantic core, stemming improves the computational efficiency and accuracy of algorithms like Support Vector Machine (SVM).

2.3.4 *Stopword:* The term "stopword" was first introduced by H.P. Luhn in 1958. In the field of Natural Language Processing (NLP), stopwords refer to commonly used words that are typically excluded from indexing or searching by computer systems. Examples of stopwords include "a," "the," and "is." Removing stopwords is a common preprocessing step in many NLP applications, such as Information Retrieval (IR) and Text Classification (TC). The benefits of stopword removal include reducing the corpus size by 35–45%, enhancing the efficiency and accuracy of text mining processes, and minimizing the time and computational complexity of the overall application. This paper examines various techniques developed over the years for identifying stopwords, applicable to both Indian and non-Indian languages. It also reviews methods for generating stopword lists along with their specific characteristics and evaluates the impact of stopword removal techniques on TC and IR applications. Additionally, it provides a detailed resource list of publicly available static stopword datasets for different languages, offering a quick reference for researchers and practitioners [11].

Stopwords are common words that do not carry significant meaning in the context of sentiment analysis. Examples include "and," "or," "the," "is," and "at." Removing these words reduces noise in the dataset and ensures that the algorithm focuses on terms that contribute to sentiment classification [12]. Stopword removal is a key preprocessing step, particularly for large text datasets, as it minimizes redundancy and allows the model to prioritize words that influence sentiment or context.

In this research, a predefined list of stopwords was used, typically sourced from libraries such as NLTK or spaCy. These libraries offer language-specific stopword lists that include high-frequency, low-importance words commonly found in text data. Domain-specific adjustments were made to customize the stopword list further. For instance, terms like "Spotify," "music," or "playlist" were retained even if they appeared frequently, as they are central to the analysis of user reviews. Conversely, words deemed irrelevant to the analysis were added to the stopword list to ensure comprehensive removal of noise.

The removal of stopwords simplifies the dataset by significantly reducing its dimensionality. This step enhances the efficiency of machine learning models by reducing computational overhead, especially in algorithms like Support Vector Machine (SVM) that process high-dimensional data. Moreover, stopword removal works synergistically with other preprocessing methods such as stemming and tokenization. By eliminating irrelevant words, the dataset becomes more focused, enabling better feature extraction and improving the accuracy of sentiment classification [13]. This step is crucial in text-heavy tasks, ensuring that the model analyzes meaningful terms and avoids being overwhelmed by noise.

2.4 TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) is a widely used technique for text representation in natural language processing tasks. In the context of analyzing Spotify user reviews, TF-IDF helps transform unstructured text data into numerical vectors that machine learning algorithms, such as Support Vector Machine (SVM), can process effectively. By quantifying the relevance of terms within and across documents, TF-IDF captures the significance of specific words concerning their occurrence patterns [14].

TF-IDF works by balancing two metrics: term frequency (TF) and inverse document frequency (IDF). TF evaluates how often a term appears within a single document, emphasizing words that are important in that specific context. Meanwhile, IDF diminishes the influence of terms that are ubiquitous across all documents, such as common stopwords, by assigning them a lower weight. For example, terms like "recommendation" and "audio quality" in Spotify reviews might receive higher TF-IDF scores because they are frequent in certain reviews but not in the entire corpus. This balance ensures that the algorithm identifies contextually significant terms while filtering out less informative ones.

Indonesian sentiment analysis has recently become a prominent area of research, particularly focusing on Twitter data. Various methods have been employed to classify sentiments expressed on Twitter, with machine learning being among the most effective. Support Vector Machine (SVM) is a widely recognized machine learning algorithm known for its robust performance. However, research on SVM has yielded varying results, with some findings showing no significant advantage over other algorithms. These discrepancies often arise from differences in SVM configurations used in each study. Kernel functions play a critical role in enhancing SVM performance, and selecting an appropriate kernel function is vital for its specific application. To address this issue, this study compares the performance of four SVM kernel functions: Polynomial, Sigmoid, Linear, and Radial Basis Function (RBF) using TF-IDF as a feature extraction and selection technique to optimize SVM's effectiveness [15].

As web technologies continue to advance, aspect-based sentiment analysis (ABSA) has gained significant attention as a method for identifying and extracting user opinions. Based on the characteristics of user comments, the ABSA problem is reframed as a multi-label classification task. A classification model utilizing BERT combined with a modified TF-IDF approach is proposed to address this challenge. In the feature extraction process, the modified TF-IDF method is designed to better capture the importance of words within individual classes by assigning distinct weights. Additionally, new features are generated by integrating BERT embeddings with the modified TF-IDF, which are then passed into a fully connected layer to fine-tune the BERT model specifically for multi-label classification tasks [16].

In this study, TF-IDF plays a critical role in feature extraction by focusing on the most relevant terms for sentiment analysis. The resulting numerical vectors are fed into the optimized SVM model, enabling it to classify reviews as positive, neutral, or negative. TF-IDF ensures that the model prioritizes words that contribute most to understanding user sentiment, improving its accuracy and efficiency. Moreover, TF-IDF is computationally efficient, making it well-suited for large datasets like Spotify reviews. Its ability to represent textual data as weighted numerical features allows for deeper insights into user feedback, aiding in the identification of patterns and trends that would otherwise remain hidden.

2.5 Validation and Evaluation

Validation and evaluation are critical steps to ensure the reliability and performance of the Support Vector Machine (SVM) model in analyzing Spotify user reviews. Validation involves testing the model's ability to generalize to unseen data, while evaluation measures its accuracy and effectiveness in classifying sentiments. In this study, 10-fold cross-validation is employed, where the dataset is split into ten parts. Nine parts are used for training, and one part is used for testing, iteratively covering all subsets. This method ensures a robust evaluation by reducing the bias caused by random data splits and provides a comprehensive measure of model performance across different data partitions.

The choice of evaluation metrics is crucial in sentiment analysis tasks, especially when dealing with imbalanced datasets where some sentiment classes, such as negative reviews, are underrepresented. Accuracy, a straightforward metric, indicates the overall proportion of correctly predicted reviews but may not fully reflect model performance in such cases. Precision and recall, on the other hand, provide a more detailed understanding. Precision measures the proportion of true positive predictions within all predicted positives, highlighting the model's reliability in identifying specific sentiments. Recall, or sensitivity, assesses the model's ability to detect all true positive instances of sentiment. The F1-score, combining precision and recall into a single value, is particularly valuable for datasets with class imbalances, offering a balanced evaluation of the model's performance.

In addition to these metrics, a confusion matrix is employed to provide a detailed visualization of the model's predictions for each sentiment class. The matrix highlights areas where the model performs well and identifies misclassifications, offering actionable insights for optimization. Furthermore, the study compares multiple optimization techniques, including Forward Selection, Backward Elimination, and AdaBoost, to determine the most effective method for enhancing the SVM model's performance. This comparative analysis reveals which techniques best address specific challenges, such as class imbalance or feature selection, ensuring that the final model is both accurate and efficient.

2.6 Training Data Modelling

Machine learning models often perform optimally when datasets are meticulously prepared and split by domain experts. Transforming real-world datasets into structured training, testing, and validation subsets demands significant effort, often involving repeated random splitting to achieve satisfactory evaluation metrics. To address the complexities of random splitting, an algorithmic approach has been introduced, enabling evenly distributed and representative dataset splits in a systematic and standardized manner [17]. The accuracy of machine learning predictions can differ depending on the various combinations of training and testing datasets. Experiments are typically conducted using different data combinations, and the Mean Absolute Error is computed by comparing the actual values from the test data with the predicted values generated by the model [18].

Training data modeling is a crucial step in building and evaluating a machine learning model. In this study, the dataset of Spotify user reviews is split into 80% training data and 20% testing data, a common practice to ensure sufficient data for training while reserving a portion for testing the model's generalization ability. This split provides the model with a comprehensive range of examples during training while preserving the integrity of the testing phase by using unseen data for evaluation.

To further enhance the effectiveness of training, techniques such as stratified sampling are employed to maintain the class distribution across the training and testing datasets. This ensures that minority sentiment classes, such as negative reviews, are adequately represented during model training and evaluation. Additionally, data augmentation methods can be applied to the training set to address imbalances by artificially increasing the representation of underrepresented classes. These steps are vital for enabling the model to learn effectively across all sentiment categories.

The training process involves feeding preprocessed data (TF-IDF vectors) and sentiment labels into the Support Vector Machine (SVM) model. The SVM identifies patterns and relationships within the dataset by learning an optimal hyperplane that separates sentiment classes. Hyperparameter tuning is often integrated during this stage to optimize key parameters such as the kernel type and regularization constant CCC, which significantly influence the model's performance. Techniques such as grid search or random search are commonly used to identify the best combination of these parameters, ensuring that the model achieves both high accuracy and generalization.

The testing set (20%) is used to evaluate the model's robustness and performance on unseen data. Metrics such as accuracy, precision, recall, and F1-score are computed to provide a comprehensive assessment of the model's effectiveness. To further validate the results, advanced evaluation techniques such as cross-validation or bootstrapping can be employed, providing a more robust assessment of the model's performance. These evaluations help ensure that the model is not only accurate on the training data but also performs well on new, real-world data, making it a reliable tool for sentiment classification.

2.7 Model Optimization

Model optimization is a vital process to enhance the performance of the SVM model in classifying Spotify user reviews. The optimization phase focuses on fine-tuning the model to achieve higher accuracy, better precision, and efficiency in handling complex datasets. This study employs several optimization techniques, such as Forward Selection, Backwards Elimination, Optimized Selection, Bagging, and AdaBoost, to improve the model’s ability to identify patterns in user feedback.

Forward Selection is a feature selection technique where relevant attributes are added incrementally to the model. Starting with an empty set of features, the algorithm iteratively evaluates and includes the features that contribute most significantly to the model's performance. This approach is particularly useful in reducing computational complexity while retaining the most informative features, such as key terms from Spotify reviews related to playlists, audio quality, or pricing.

Forward Selection SVM is a method that combines the SVM algorithm with the Forward Selection (FS) feature selection technique. The main goal of this approach is to improve the performance of the SVM model by selecting the most relevant features [19]. The Forward Selection SVM method involves two main steps: applying the SVM algorithm with different kernel functions (e.g., dot, polynomial, RBF) to the dataset and applying the Forward Selection feature selection technique to the SVM model to identify the most important features and further improve the model's accuracy. The studies show that the SVM with RBF kernel combined with Forward Selection (SVM(RBF)+FS) outperforms other SVM models and machine learning algorithms in predicting datasets such as breast cancer and chronic kidney disease. Additionally, the Forward Selection SVM method has been found to reduce the training and testing time of the SVM model while also improving its recognition accuracy [20].

In contrast, Backwards Elimination begins with the full set of features and systematically removes the least significant ones. This method ensures that irrelevant or redundant attributes are excluded from the dataset, making the model more efficient. By focusing on high-impact features, Backwards Elimination helps the model maintain a strong predictive performance with fewer input dimensions.

Backwards Elimination is a feature selection technique that identifies and removes irrelevant features using a linear regression model as a basis. This method aims to select the most relevant features for the model, enhancing its overall efficiency. Among its advantages are reduced training time, decreased model complexity, and improved performance and accuracy. Consequently, the application of Backwards Elimination contributes to enhancing the performance of the SVM algorithm [21]. The use of the Backwards Elimination method can enhance the accuracy of the SVM algorithm while effectively selecting relevant attributes or variables. By reducing the number of features from 30 to 13, this method significantly improves the performance of both classification models, optimizing their efficiency and predictive capabilities [22].

Optimized Selection combines the strengths of Forward Selection and Backwards Elimination to evaluate the importance of each feature comprehensively. This hybrid approach identifies the optimal subset of features, balancing accuracy and efficiency. By iteratively adding significant features (as in Forward Selection) and removing irrelevant ones (as in Backwards Elimination), Optimized Selection ensures a refined feature set that enhances model performance. The application of this method in this study allows the Support Vector Machine (SVM) model to focus on critical attributes in Spotify user reviews while minimizing computational overhead.

One of the key advantages of Optimized Selection is its ability to mitigate the effects of noisy or redundant features that can degrade model performance. By eliminating unnecessary attributes, the approach reduces the dimensionality of the dataset, leading to faster training times and improved generalization. In the context of sentiment analysis, Optimized Selection ensures that the model prioritizes terms and patterns most indicative of sentiment, such as keywords related to user satisfaction or dissatisfaction, rather than irrelevant or repetitive content [23].

This method also enhances the interpretability of the SVM model. By narrowing the feature set to only the most impactful attributes, the study provides clearer insights into the factors driving user sentiment. For example, in analyzing Spotify reviews, Optimized Selection might highlight terms like "recommendation quality" or "subscription cost" as pivotal features. These insights not only improve the model's classification accuracy but also offer actionable information for decision-makers looking to refine Spotify's services based on user feedback.

Beyond feature selection, ensemble methods like Bagging are employed to improve the stability of the model. Bagging involves training multiple SVM models on different subsets of the training data and combining their predictions. By aggregating predictions, Bagging reduces the variance inherent in individual models and enhances the overall robustness of the system. This ensemble approach is particularly effective in handling diverse user sentiments, as it minimizes the risk of overfitting to specific data samples and ensures more generalized performance [24].

One of the primary strengths of Bagging is its ability to mitigate the effects of noisy or imbalanced datasets. By creating multiple bootstrapped subsets of the training data, Bagging ensures that each model is trained on a slightly different perspective of the dataset. This diversity among models makes the ensemble less sensitive to outliers and noise, which are common in large-scale datasets like Spotify user reviews. The final prediction is typically determined through majority voting or averaging, providing a more reliable outcome compared to individual models [25].

In sentiment analysis tasks, Bagging proves particularly valuable when dealing with underrepresented sentiment classes, such as negative reviews. By training multiple models on varied subsets, Bagging increases the likelihood that minority class examples are adequately represented across the ensemble. This improves the model's ability to classify sentiments across all categories accurately. Moreover, Bagging complements other optimization techniques, such as feature selection and hyperparameter tuning, to create a robust analytical framework for handling complex and imbalanced datasets [26].

Another ensemble method, AdaBoost, focuses on addressing the model's weaknesses by iteratively reweighting the importance of misclassified reviews. This method works by assigning higher weights to reviews that are challenging to classify correctly, compelling the SVM model to focus more on these difficult cases [27]. By doing so, AdaBoost ensures that the model incrementally improves its accuracy over successive iterations, making it a powerful tool for handling imbalanced datasets and nuanced sentiment classes.

A key advantage of AdaBoost lies in its ability to improve the classification of minority classes, such as negative sentiments, which are often underrepresented in datasets like Spotify user reviews. In each iteration, AdaBoost adjusts the distribution of weights to emphasize misclassified examples, such as negative reviews miscategorized as neutral or positive. This iterative reweighting enables the ensemble to adapt and refine its performance, ultimately achieving a better balance between accuracy and sensitivity across all sentiment categories [28].

Moreover, AdaBoost complements other techniques like feature selection and data preprocessing by enhancing the overall robustness of the model. Its adaptive nature makes it particularly effective in addressing data imbalance and noise. However, AdaBoost's performance can be sensitive to outliers, as it assigns high importance to these samples, potentially leading to overfitting [29]. To mitigate this, AdaBoost is often combined with regularization techniques or used alongside robust preprocessing methods, ensuring a balance between adaptability and generalization. In this study, AdaBoost plays a pivotal role in refining the SVM model, enabling it to classify sentiments more accurately, particularly for challenging cases.

The optimization process also involves hyperparameter tuning to adjust critical SVM parameters, such as the kernel type, regularization parameter C, and kernel coefficient gamma. These parameters significantly influence the model's ability to generalize well on new data. Grid Search or Random Search methods are applied to systematically identify the best combination of these parameters.

Each optimization technique is evaluated using metrics such as accuracy, precision, recall, and F1-score. Additionally, cross-validation ensures that the model's performance is consistent across different subsets of the data. By comparing the results from various optimization methods, the study identifies the approach that offers the best balance between accuracy and computational efficiency.

The results of the optimization are visualized using tools like confusion matrices and performance graphs. These visualizations provide insights into the strengths and weaknesses of each optimized model, highlighting areas for improvement. For instance, while some techniques might excel in identifying positive sentiments, others may perform better in classifying negative or neutral reviews.

Overall, the optimization phase is essential for refining the SVM model to achieve superior performance in classifying Spotify user reviews. By integrating feature selection, ensemble methods, and hyperparameter tuning, the study ensures that the final model is both accurate and efficient, capable of providing actionable insights from user feedback.

3 RESULT AND DISCUSSION

The SVM model, applied to the Spotify user reviews dataset, demonstrated varying levels of performance depending on the optimization technique used.

3.1 Result using Support Vector Machine Algorithm

The baseline performance of the Support Vector Machine (SVM) model in analyzing Spotify user reviews was evaluated based on accuracy, precision, recall, and F1-score. The overall accuracy of the model was 83.5%, indicating that the SVM successfully classified the majority of user reviews correctly. However, a deeper analysis of the metrics for each sentiment class revealed important insights into the model's strengths and weaknesses. The results are in Table 1 and Figure 2.

Table 1. Result using Support Vector Machine Algorithm

	Precision	*Recall*	*F1-Score*
Negative	0.60	0.31	0.41
Neutral	0.00	0.00	0.00
Positive	0.85	0.98	0.91
Macro Avg	0.48	0.43	0.44
Weighted Avg	0.77	0.83	0.79
Accuracy :	**0.835**		

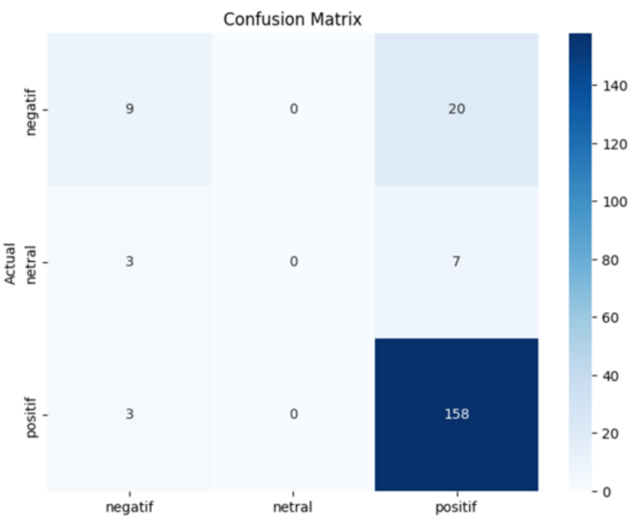

Figure 2. Confusion matrix of SVM

For positive sentiments, which had the largest representation in the dataset, the SVM performed exceptionally well. It achieved a precision of 85%, a recall of 98%, and an F1-score of 91%, highlighting the model's ability to accurately classify most positive reviews. The high recall indicates that nearly all positive reviews were identified, which aligns with the sentiment's dominance in the dataset.

In contrast, the performance for negative sentiments was considerably lower. The model achieved a precision of 60%, a recall of 31%, and an F1-score of 41%, suggesting difficulty in identifying and classifying negative reviews. Similarly, for neutral sentiments, the SVM struggled, with a precision, recall, and F1-score of 0%, indicating that it was unable to accurately detect or classify reviews in this category. These results are reflected in the macro average metrics (precision: 48%, recall: 43%, F1-score: 44%) and indicate that the model's performance is skewed towards classes with higher representation, such as positive sentiments.

The weighted average metrics (precision: 77%, recall: 83%, F1-score: 79%) account for class imbalances, providing a more balanced evaluation of the model's overall performance. While the SVM algorithm performs well on dominant classes, its inability to accurately classify minority classes like negative and neutral sentiments highlights the need for optimization techniques or methods to address class imbalances in the dataset.

The correlation heatmap of TF-IDF features highlights the relationships between terms extracted from the user reviews. The sparsity observed in the heatmap, with most term pairs showing near-zero correlation, indicates minimal redundancy among the features. This suggests that the TF-IDF vectorization effectively captures distinct contributions of each term, ensuring the dataset's features are independent and non-overlapping. The diagonal line represents the perfect correlation of each term with itself, as expected in such analyses. Figure 3 shows the heatmap.

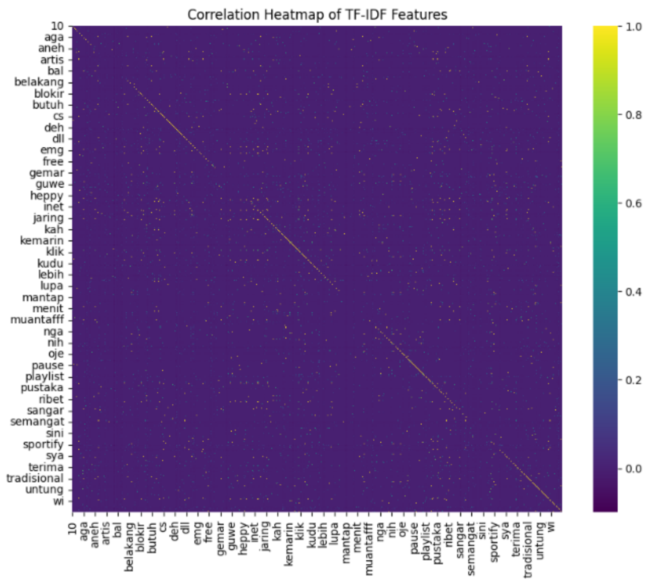

Figure 3. Heatmap correlation matrix

Notably, certain terms, such as "playlist," "spotify," and "mantap," exhibit isolated clusters, reflecting their frequent co-occurrence in reviews with positive sentiment. This observation aligns with the dominance of the positive sentiment class in the classification results. Moreover, the low overall correlation validates the robustness of the feature selection techniques employed, such as Forward Selection and Backwards Elimination, which targeted the most informative terms without introducing multicollinearity.

This heatmap provides visual evidence supporting the model's performance, confirming that the optimized features contributed to the SVM's ability to classify sentiments effectively while maintaining high precision for the dominant positive class. However, the sparsity also underscores challenges in identifying patterns for minority classes, which may require more sophisticated feature engineering or deep learning techniques for future research.

3.2 *Result Optimization Model*

The performance of the Support Vector Machine (SVM) model was enhanced using various optimization techniques, including Forward Selection, Backwards Elimination, Optimized Selection, Bagging, and AdaBoost. These methods were employed to refine feature selection, balance class representation, and improve the model's overall accuracy and classification metrics. Below are the results and insights for each optimization model.

3.2.1 *Result of SVM using Forward Selection:* The implementation of the Forward Selection optimization method resulted in an accuracy of 81.5%, slightly lower than the baseline SVM model. Forward Selection focuses on retaining only the most relevant features to improve classification efficiency, but its performance varied across sentiment classes. The following results are in Table 2 and Figure 4 below.

For positive sentiments, the model performed well with a precision of 83%, a recall of 98%, and an F1-score of 90%, indicating its strength in identifying and correctly classifying the majority class. However, the model struggled with minority classes.

Table 2. Result of SVM using Forward Selection

	Precision	*Recall*	*F1-Score*
Negative	0.56	0.17	0.26
Neutral	0.00	0.00	0.00
Positive	0.83	0.98	0.90
Macro Avg	0.45	0.38	0.39
Weighted Avg	0.75	0.81	0.76
Accuracy :	**0.815**		

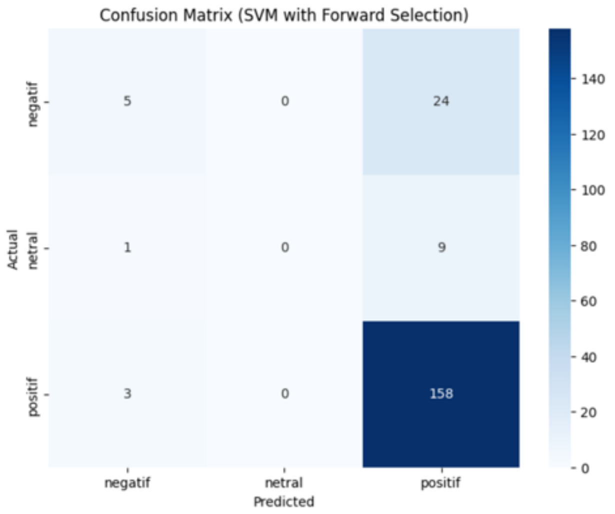

Figure 4. Confusion Matrix of SVM with FS

For negative sentiments, the model achieved a precision of 56%, a recall of 17%, and an F1-score of 26%, suggesting significant challenges in accurately identifying this sentiment class. Similarly, for neutral sentiments, the precision, recall, and F1-score were 0%, highlighting the model's inability to classify this sentiment class effectively.

The macro-average metrics (precision: 46%, recall: 38%, F1-score: 39%) reflect the overall imbalance in the model's performance across classes. The weighted average metrics (precision: 75%, recall: 81%, F1-score: 76%) were skewed toward the majority class (positive sentiments), which dominated the dataset. These results indicate that while Forward Selection refines feature selection, it may require additional enhancements, such as balancing techniques, to improve classification for minority sentiments.

3.2.2 *Result of SVM using Backwards Elimination:* The application of Backwards Elimination as an optimization technique resulted in an accuracy of 81.0%, comparable to Forward Selection. This method works by iteratively removing the least significant features, aiming to enhance the model's performance by focusing on the most impactful attributes. However, the results show varying performance across sentiment classes. The results are shown in Table 3 and Figure 5 below.

Table 3. Result of SVM using Backwards Elimination

	Precision	*Recall*	*F1-Score*
Negative	0.44	0.14	0.21
Neutral	0.00	0.00	0.00
Positive	0.83	0.98	0.90
Macro Avg	0.42	0.37	0.37

Weighted Avg	0.73	0.81	0.75
Accuracy :	**0.81**		

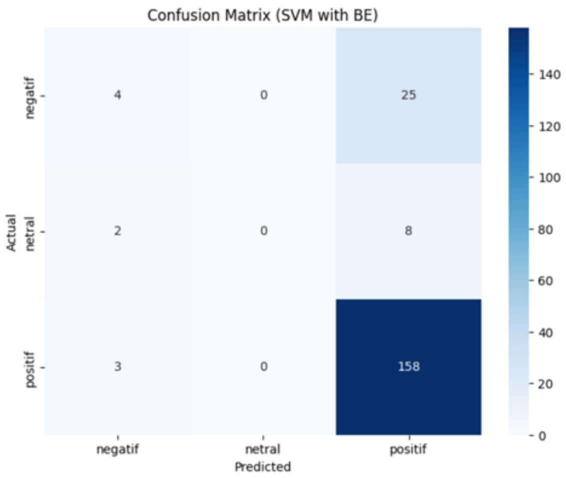

Figure 5. Confusion Matrix of SVM with BE

For positive sentiments, which dominate the dataset, the model maintained a strong performance with a precision of 83%, a recall of 98%, and an F1-score of 90%, indicating its reliability in identifying and classifying this sentiment. This class continued to benefit the most from the optimization.

The performance for negative sentiments was less satisfactory, with a precision of 44%, a recall of 14%, and an F1-score of 21%, reflecting difficulty in recognizing minority class examples. For neutral sentiments, the metrics (precision, recall, F1-score) remained at 0%, demonstrating that the model struggled to identify this category.

The macro-average metrics (precision: 42%, recall: 37%, F1-score: 37%) indicate imbalanced performance across classes, while the weighted average metrics (precision: 73%, recall: 81%, F1-score: 75%) were heavily influenced by the majority class. These results suggest that while Backwards Elimination can simplify the model by removing less relevant features, additional strategies, such as addressing class imbalance or incorporating advanced feature selection methods, are needed to improve the classification of minority sentiments.

3.2.3 *Result of SVM using Optimized Selection:* The Optimized Selection method, combining Forward Selection and Backwards Elimination, achieved the highest accuracy of 84.5% among the optimization techniques evaluated. This approach focused on both adding significant features and removing irrelevant ones, resulting in a balanced model that leveraged the strengths of both methods. The improvement in accuracy reflects its ability to refine the dataset effectively, ensuring better feature relevance and model performance. The results are shown in Table 4 below and Figure 6.

Table 4. Result of SVM using Optimized Selection

	Precision	***Recall***	***F1-Score***
Negative	0.63	0.41	0.50
Neutral	0.25	0.10	0.14
Positive	0.88	0.97	0.92
Macro Avg	0.59	0.49	0.52
Weighted Avg	0.81	0.84	0.82
Accuracy :	**0.845**		

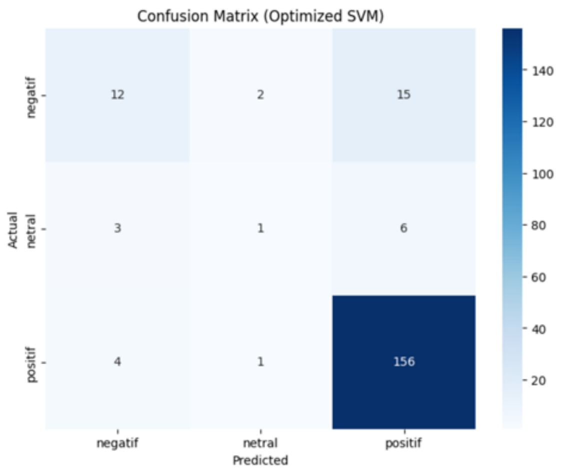

Figure 6. Confusion Matrix of SVM With OS

For positive sentiments, the model performed exceptionally well with a precision of 88%, a recall of 97%, and an F1-score of 92%, confirming its reliability in handling the majority class. The high recall indicates that nearly all positive reviews were correctly identified, and the balance between precision and recall demonstrates the model's efficiency in reducing false positives.

The performance for negative sentiments showed noticeable improvement compared to other methods, with a precision of 63%, a recall of 41%, and an F1-score of 50%. These results indicate the model's better capability to identify negative reviews, though challenges remain in capturing a larger portion of the examples in this minority class.

The model struggled with neutral sentiments, , achieving a precision of 25%, a recall of 10%, and an F1-score of 14%. While the inclusion of features relevant to neutrality contributed slightly to the results, this category remained difficult to classify due to the limited representation in the dataset.

The macro-average metrics (precision: 59%, recall: 49%, F1-score: 52%) show an overall improvement in balance across classes compared to Forward and Backwards Elimination. Meanwhile, the weighted averages (precision: 81%, recall: 84%, F1-score: 82%) highlight the model's strong performance on the dominant positive class. These findings suggest that Optimized Selection effectively refines the feature set and improves classification, making it the most robust optimization method for this study. However, additional techniques, such as oversampling or advanced embeddings, may further enhance the model's performance on underrepresented classes.

3.2.4 *Result of SVM using Bagging:* The Bagging (Bootstrap Aggregating) technique applied to the Support Vector Machine (SVM) model achieved an accuracy of 82.5%, reflecting its effectiveness in improving model stability. Bagging works by generating multiple subsets of the dataset through random sampling with replacement and training individual SVM classifiers on each subset. The final prediction is obtained by aggregating the outputs of these classifiers, typically through majority voting. The results are shown in Table 5 and Figure 7 below. For the positive sentiment class, Bagging SVM performed strongly, with a precision of 85%, a recall of 98%, and an F1-score of 91%. These metrics highlight the model's ability to correctly classify most positive reviews while maintaining a low false positive rate. The high recall score indicates that nearly all positive reviews in the test set were identified, making Bagging particularly effective for the dominant class.

Table 5. Result of SVM using Bagging

	Precision	*Recall*	*F1-Score*
Negative	0.53	0.28	0.36
Neutral	0.00	0.00	0.00
Positive	0.85	0.98	0.91
Macro Avg	0.46	0.42	0.42
Weighted Avg	0.76	0.82	0.78
Accuracy	**: 0.825**		

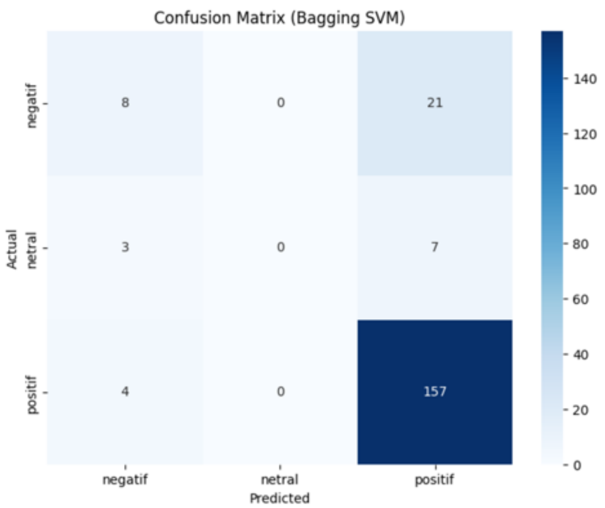

Figure 7. Confusion Matrix of SVM with Bagging

In the case of negative sentiments, the model exhibited moderate performance with a precision of 53%, a recall of 28%, and an F1-score of 36%. While these metrics indicate some improvement in recognizing negative reviews compared to the baseline, the recall score remains relatively low. This suggests that the Bagging approach was only partially successful in addressing the minority class representation issue.

The model struggled significantly with neutral sentiments, where precision, recall, and F1-score were all 0%. This indicates that the model was unable to classify neutral reviews effectively, a challenge consistent across all optimization techniques. The limited number of neutral reviews in the dataset likely contributed to this poor performance, as the training process may not have provided sufficient exposure to this class. The macro-average metrics (precision: 46%, recall: 42%, F1-score: 42%) indicate an overall imbalance in the model's ability to classify all sentiment classes equally. These averages are skewed due to the underperformance of negative and neutral sentiments, highlighting a potential bias in the model's ability to accurately classify these categories. However, the weighted averages (precision: 76%, recall: 82%, F1-score: 78%) reflect better performance, as they are influenced by the dominance of the positive sentiment class in the dataset.

Bagging SVM demonstrated its strengths in enhancing classification stability and reducing overfitting, particularly for the positive sentiment class. The ensemble approach allowed the model to generalize better, especially for high-representation classes, but it had limited success in handling underrepresented sentiments.

One possible reason for Bagging's limited improvement in minority classes is the lack of diversity in the resampled datasets. While Bagging creates multiple subsets, the class imbalance in the original dataset likely persisted in most of these subsets, hindering the model's ability to learn patterns for negative and neutral sentiments.

Overall, Bagging proved to be a useful optimization technique for improving the stability and accuracy of SVM, particularly for the dominant positive sentiment class. However, its limited effectiveness in handling minority classes underscores the need for further refinements to enhance its utility in imbalanced datasets like Spotify user reviews.

The findings suggest that while Bagging enhances model robustness, its ability to improve minority class classification depends on additional preprocessing and sampling strategies. This makes it a promising but not standalone solution for sentiment analysis tasks involving imbalanced data.

Result of SVM using Adaboost: The AdaBoost (Adaptive Boosting) technique applied to the SVM model achieved an overall accuracy of 80.5%.

AdaBoost works by sequentially training weak classifiers and assigning higher weights to misclassified samples in subsequent iterations, aiming to improve the model's ability to handle challenging cases. While this approach enhances the classification of dominant classes, its performance on minority classes remained limited in this study. The following results are in Table 6 and Figure 8 below.

For the positive sentiment class, which dominated the dataset, AdaBoost achieved a precision of 81%, a recall of 100%, and an F1-score of 89%. The perfect recall score indicates that AdaBoost successfully identified all positive reviews in the test set. However, the precision score reflects the occurrence of some false positives, where reviews from other classes were misclassified as positive.

In contrast, the model's performance for negative sentiments was poor, with precision, recall, and F1-score all at 0%. Similarly, for neutral sentiments, the metrics remained at 0%, indicating that AdaBoost was unable to correctly classify any reviews from these classes. This result suggests that the boosting mechanism disproportionately focused on the dominant positive class, neglecting the underrepresented negative and neutral classes.

Table 6. Result SVM using AdaBoost

	Precision	*Recall*	*F1-Score*
Negative	0.00	0.00	0.00
Neutral	0.00	0.00	0.00
Positive	0.81	1.00	0.89
Macro Avg	0.27	0.33	0.30
Weighted Avg	0.65	0.81	0.72
Accuracy :	**0.825**		

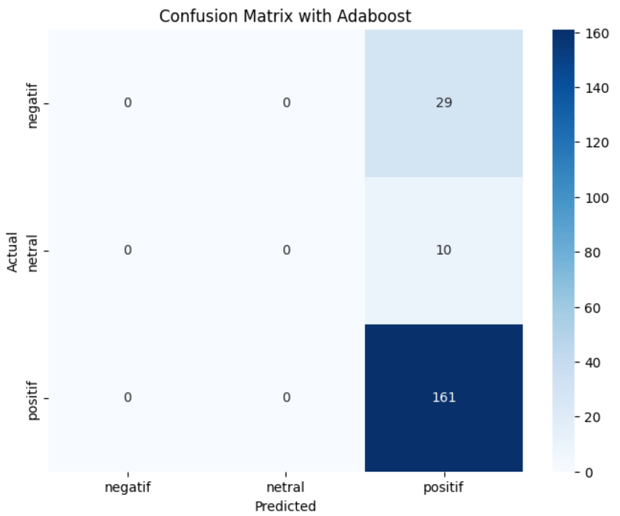

Figure 8. Confusion Matrix of SVM With Adaboost

The macro-average metrics (precision: 27%, recall: 33%, F1-score: 30%) highlight the imbalance in the model’s classification performance across all sentiment classes. The low macro-average precision and recall scores underscore the model's inability to generalize well for minority classes. However, the weighted averages (precision: 65%, recall: 81%, F1-score: 72%) were higher, reflecting the positive class's dominance in the dataset and the model's strength in classifying it accurately.

AdaBoost’s reliance on iterative reweighting of misclassified samples appears to have favored the positive sentiment class due to its higher representation in the dataset. This imbalance in focus led to significant misclassifications for the negative and neutral classes, resulting in poor performance for these categories.

One limitation of AdaBoost in this context is its sensitivity to noise or imbalanced data. The small number of negative and neutral samples in the training set likely caused the algorithm to assign insufficient attention to these classes during the boosting process. This issue could be mitigated by incorporating additional preprocessing techniques, such as oversampling or under-sampling, to balance the dataset.

Another potential enhancement for AdaBoost is the integration of feature engineering techniques tailored to minority classes. By identifying and emphasizing features unique to negative and neutral reviews, the model could improve its ability to classify these sentiments effectively.

Despite its limitations, AdaBoost demonstrated its ability to refine the classification of the majority class, showcasing its potential in handling dominant categories. However, its application in highly imbalanced datasets, such as Spotify user reviews, requires supplementary strategies to address the unequal distribution of sentiment classes.

In conclusion, while AdaBoost improved the classification of positive sentiments, its inability to handle minority classes underscores the importance of balancing datasets and refining boosting mechanisms. Future research should focus on hybrid approaches that combine boosting with data balancing techniques to enhance overall performance across all sentiment classes.

3.3 Comparison of Optimization Model

The comparison of optimization models for the SVM (Support Vector Machine) algorithm reveals the impact of various techniques on model accuracy. Figure 9 shows the accuracy performance of the baseline SVM model and the optimized models using different methods, such as feature selection (FS), backwards elimination (BE), oversampling (OS), and others. The results highlight the effectiveness of each optimization method in enhancing the predictive capability of the SVM algorithm.

As depicted in Figure 9, the oversampling (OS) method achieved the highest accuracy of 0.845, outperforming the baseline SVM model (0.835). In contrast, other methods, such as feature selection (FS) and backwards elimination (BE), showed moderate improvements, with accuracy values of 0.815 and 0.810, respectively. The SVM + A model recorded the lowest accuracy of 0.805. These findings emphasize the critical role of oversampling in addressing class imbalance and boosting the SVM model's performance compared to other optimization strategies.

3.4 Result of Data Visualization

The analysis of user reviews provides valuable insights into the perception of the Spotify application. To better understand the key themes and sentiments expressed by users, a word cloud visualization was generated, as shown in Figure 10. This visualization highlights the most frequently occurring words in the dataset, offering a concise summary of user feedback.

As illustrated in Figure 10, the most prominent words include "bagus","lagu" and "premium," which indicate a strong positive sentiment among users regarding song quality and premium features. Meanwhile, terms like "iklan" suggest a recurring topic of discussion, possibly reflecting user concerns or mixed opinions about advertisements. This visualization underscores the key areas where Spotify excels and points to aspects that may warrant further attention for improvement.

The results of this study reveal key patterns in Spotify user reviews, with the positive sentiment class being the most dominant, comprising a significant majority of the dataset. This dominance is likely attributable to user satisfaction with Spotify's personalized playlists, intuitive interface, and audio quality, which are frequently highlighted in the reviews. These positive patterns were identified using the SVM model with features extracted through TF-IDF, focusing on terms such as "great," "recommendation," and "premium."

The SVM model, optimized with methods such as Forward Selection, Backwards Elimination, and Bagging, achieved its highest performance (84.5% accuracy) using Optimized Selection. This technique effectively identified key terms associated with positive sentiments, resulting in high precision (88%) and recall (97%) for the positive class. These metrics demonstrate the model's capability to capture and classify dominant patterns accurately.

In contrast, the minority sentiment classes (neutral and negative) presented significant challenges. Negative sentiments, often associated with dissatisfaction regarding pricing, advertisements, or recommendation algorithms, were underrepresented in the dataset, resulting in lower precision (63%) and recall (41%). Neutral sentiments exhibited the weakest performance, with precision, recall, and F1-score near zero due to their minimal representation and ambiguous language.

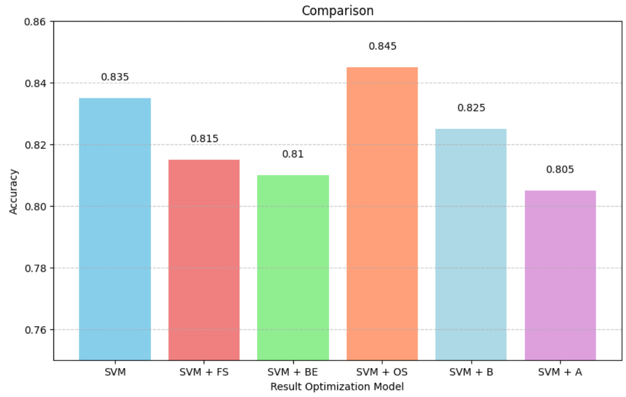

Figure 9. Data visualization

Figure 10. Data visualization

The numerical findings highlight the need for advanced techniques to address class imbalance, particularly for identifying minority sentiment patterns. Future research integrating deep learning approaches could enhance the representation and classification of these classes, providing a more balanced analysis of user sentiment patterns. These insights reaffirm the importance of combining data-driven optimization techniques with advanced analytical methods to capture both dominant and nuanced sentiment trends in user reviews.

The sentiment patterns found in this study indicate that positive reviews dominate, primarily concerning the quality of playlist recommendations and audio experience. Meanwhile, negative reviews are mainly associated with complaints about advertisements and subscription pricing. The SVM model faced challenges in classifying negative and neutral reviews due to class imbalance in the dataset, which was addressed through optimization and data augmentation techniques.

3.5 Discussion

The results of this study indicate that the Support Vector Machine (SVM) algorithm, optimized with various feature selection techniques and ensemble methods, successfully improved sentiment classification accuracy for Spotify user reviews. The model using the Optimized Selection method achieved the highest accuracy of 84.5%, outperforming other methods such as Forward Selection, Backwards Elimination, Bagging, and AdaBoost.

Comparing these findings with previous research, W. M. P.D. and Haryoko [30] utilized Genetic Algorithm to optimize SVM parameters for sentiment analysis of Go-Jek service reviews, demonstrating a significant increase in classification accuracy. This aligns with the findings of this research, emphasizing the crucial role of algorithm optimization in enhancing classification accuracy, particularly in handling class imbalance within datasets.

Additionally, Sharazita Dyah Anggita and Ferian Fauzi Abdulloh [31] revealed that combining Particle Swarm Optimization (PSO) with Information Gain improved sentiment analysis accuracy up to 95%, especially in legal document classification. This suggests that metaheuristic-based optimization techniques could also serve as a potential alternative to further enhance classification performance in complex and diverse datasets.

However, this study still faces challenges, particularly in classifying neutral and negative sentiments. While Optimized Selection improved negative sentiment classification to 41% recall and 63% precision, the neutral sentiment category still exhibited low classification accuracy. This indicates that feature selection and ensemble-based optimization alone do not fully address class imbalance issues in this dataset.

As a solution, future studies may consider further data augmentation techniques, such as the Synthetic Minority Oversampling Technique (SMOTE) or word embeddings, to improve minority class representation. Additionally, deep learning-based approaches, such as Convolutional Neural Network (CNN) or Bidirectional Long Short-Term Memory (BiLSTM), could be explored to capture more complex patterns in user review texts.

Overall, this study contributes to the development of a more balanced and accurate SVM-based sentiment analysis model while opening avenues for future research into more sophisticated optimization techniques to handle large-scale user review data.

4 CONCLUSION

This study demonstrates the effectiveness of employing optimization techniques to enhance the performance of the SVM algorithm in analyzing user reviews. The integration of methods such as Forward Selection, Backward Elimination, Bagging, and AdaBoost highlights their respective contributions in refining feature selection, addressing class imbalance, and improving classification accuracy. Among these techniques, Optimized Selection achieved the highest performance, emphasizing the importance of combining feature addition and removal strategies. While the SVM model performed well for the dominant sentiment class, challenges remain in accurately classifying minority classes like neutral and negative reviews. These findings underscore the need for continued exploration of hybrid approaches and advanced preprocessing methods to further enhance model robustness.

Additionally, this research contributes to the development of pattern analysis and sentiment classification models by addressing class imbalance and optimizing performance metrics. Future studies could explore the integration of deep learning techniques, such as Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN), to improve the classification of minority classes. These models, known for their ability to capture complex patterns and non-linear relationships, could complement current methods and provide significant improvements. By advancing classification balance and accuracy, this research lays the groundwork for more effective sentiment analysis in user-driven platforms like Spotify, with broader implications for various industries reliant on user feedback.

AUTHOR'S CONTRIBUTION

The authors collaboratively contributed to the completion of this research. Wakhid Kurniwan led the conceptualization and development of the research framework, conducted comprehensive data analysis, and was primarily responsible for drafting the manuscript. Nova Tri Romadloni provided significant input in designing the methodological approach, conducting an extensive review of relevant literature, and critically evaluating the manuscript to ensure academic rigor. Both authors engaged in thorough discussions during the research process, contributed to the interpretation of findings, and approved the final manuscript for submission.

COMPETING INTERESTS

The authors declare that there are no competing interests or conflicts of interest (COI) associated with this publication. All aspects of the research, including its design, execution, and reporting, were conducted impartially and independently.

ACKNOWLEDGMENT

The authors extend their heartfelt appreciation to everyone who provided valuable insights, moral and material support, constructive feedback, and assistance in various forms, all of which were instrumental in the successful completion of this study.

REFERENCES

[1] D.-A. Joshi and B. Patel, “Data Preprocessing: The techniques for preparing clean and quality data for data analytics process,” *Orient. J. Comput. Sci. Technol.*, vol. 13, pp. 78–81, Jan. 2021, doi: 10.13005/ojcst13.0203.03.

[2] B. Malley, D. Ramazzotti, and J. T. Wu, “Data Prp-processing BT - Secondary analysis of electronic health records,” M. I. T. C. Data, Ed., Cham: Springer International Publishing, 2016, pp. 115–141. doi: 10.1007/978-3-319-43742-2_12.

[3] R. A. Sayany and F. Karachi, “Overview of Data Cleansing Process in Data Mining,” June, 2021, doi: 10.13140/RG.2.2.22323.76326.

[4] M. Sathiyanarayanan, A. Junejo, and O. Fadahunsi, *Visual Analysis of Predictive Policing to Improve Crime Investigation*. 2019. doi: 10.1109/IC3I46837.2019.9055515.

[5] M. R, “Natural language processing for analysing and extracting insights,” *International J. Sci. Res. Eng. Manag.*, vol. 06, no. 06, pp. 1–4, 2022, doi: 10.55041/ijsrem14434.

[6] S. Y. Senanayake, K. T. P. M. Kariyawasam, and P. S. Haddela, “Enhanced tokenizer for Sinhala Language,” in *2019 National Information Technology Conference (NITC)*, 2019, pp. 84–89. doi: 10.1109/NITC48475.2019.9114420.

[7] C. G. S. Satwika and J. P. Pramod, “Exploring Word-Level

representations in modern natural language processing," International Journal for Research in Applied Science & Engineering Technology, vol. 12, no. X, pp. 21–28, October, 2024.

[8] A. Jabbar, S. Iqbal, M. I. Tamimy, S. Hussain, and A. Akhunzada, "Empirical evaluation and study of text stemming algorithms," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5559–5588, 2020, doi: 10.1007/s10462-020-09828-3.

[9] I. Mazlin, I. M. Rawi, M. Z. Zakaria, and S. S. A. Rahman, "Comparative study of stemming strategy for Hadith text classification," in *2021 3rd International Conference on Natural Language Processing (ICNLP)*, 2021, pp. 42–45. doi: 10.1109/ICNLP52887.2021.00013.

[10] N. Pamungkas *et al.*, "Comparison of stemming test results of Tala Algorithms with Nazief Adriani in abstract documents and national news," *Inf. J. Ilm. Bid. Teknol. Inf. dan Komun.*, vol. 8, no. 1 SE-Articles, pp. 33–41, Jan. 2023, doi: 10.25139/inform.v8i1.5569.

[11] D. J. Ladani and N. P. Desai, "Stopword identification and removal techniques on TC and IR applications: A Survey," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2020, pp. 466–472. doi: 10.1109/ICACCS48705.2020.9074166.

[12] H. Fang, "An Ultrabroadband Photodetector Based on PMN-28PT Single Crystal," 2020, pp. 49–73. doi: 10.1007/978-981-15-4312-8_3.

[13] A. W. Pradana and M. Hayaty, "The effect of stemming and removal of Stopwords on the accuracy of sentiment analysis on Indonesian-language texts," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 4 SE-, pp. 375–380, Oct. 2019, doi: 10.22219/kinetik.v4i4.912.

[14] F. Alzami, E. D. Udayanti, D. P. Prabowo, and R. A. Megantara, "Document preprocessing with TF-IDF to improve the Polarity Classification Performance of unstructured sentiment analysis," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 5, no. 3 SE-, pp. 235–242, Aug. 2020, doi: 10.22219/kinetik.v5i3.1066.

[15] P. H. Prastyo, I. Ardiyanto, and R. Hidayat, "Indonesian Sentiment Analysis: An experimental study of four Kernel Functions on SVM Algorithm with TF-IDF," in *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*, 2020, pp. 1–6. doi: 10.1109/ICDABI51230.2020.9325685.

[16] Z. Jin, X. Lai, and J. Cao, "Multi-label sentiment analysis Base on BERT with modified TF-IDF," in *2020 IEEE International Symposium on Product Compliance Engineering-Asia (ISPCE-CN)*, 2020, pp. 1–6. doi: 10.1109/ISPCE-CN51288.2020.9321861.

[17] I. N. Syamsuriana Basri, Fitrawahyudi, Khaerani, S. M. , Ernawati , Aryanti, S. Aisyah, and Irma Sakti, "Peningkatan kemampuan literasi digital di lingkungan pendidikan berbasis aplikasi Canva," vol. 1, no. 2, pp. 96–103, 2023, doi: 10.37985/pmsdu.v1i2.65.

[18] R. Medar, V. S. Rajpurohit, and B. Rashmi, "Impact of training and testing data splits on accuracy of Time Series Forecasting in machine learning," in *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, 2017, pp. 1–6. doi: 10.1109/ICCUBEA.2017.8463779.

[19] H.- Harafani, "Forward Selection pada Support Vector Machine untuk mMemprediksi kanker payudara," *J. Infortech*, vol. 1, no. 2, pp. 131–139, 2020, doi: 10.31294/infortech.v1i2.7398.

[20] E. Purwaningsih, "Improving the performance of Support Vector Machine with Forward Selection for Prediction of chronic kidney disease," *JITK (Jurnal Ilmu Pengetah. dan Teknol. Komputer)*, vol. 8, no. 1, pp. 18–24, 2022, doi: 10.33480/jitk.v8i1.3327.

[21] A. Bode, "Perbandingan metode prediksi Support Vector Machine dan Linear Regression menggunakan Backward Elimination pada produksi minyak kelapa," *Simtek J. Sist. Inf. dan Tek. Komput.*, vol. 4, no. 2, pp. 104–107, 2019, doi: 10.51876/simtek.v4i2.57.

[22] F. Maulidina, Z. Rustam, S. Hartini, V. V. P. Wibowo, I. Wirasati, and W. Sadewo, "Feature optimization using Backward Elimination and Support Vector Machines (SVM) algorithm for diabetes classification," *J. Phys. Conf. Ser.*, vol. 1821, no. 1, 2021, doi: 10.1088/1742-6596/1821/1/012006.

[23] N. Z. Dina, S. D. Ravana, and N. Idris, "An experimental study on Hybrid Feature Selection Techniques for Sentiment Classification," in *2022 14th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, 2022, pp. 270–275. doi: 10.1109/SKIMA57145.2022.10029452.

[24] Y. Gong, B. Liao, P. Wang, and Q. Zou, "DrugHybrid_BS: Using hybrid feature combined with Bagging-SVM to predict potentially druggable proteins," *Front. Pharmacol.*, vol. 12, no. November, pp. 1–12, 2021, doi: 10.3389/fphar.2021.771808.

[25] S. Huang and J. Zhou, "An enhanced stability evaluation system for entry-type excavations: Utilizing a hybrid bagging-SVM model, GP and kriging techniques," *J. Rock Mech. Geotech. Eng.*, 2024, doi: https://doi.org/10.1016/j.jrmge.2024.05.024.

[26] W. Fan, B. Xu, H. Li, G. Lu, and Z. Liu, "A novel surrogate model for channel geometry optimization of PEM fuel cell based on Bagging-SVM Ensemble Regression," *Int. J. Hydrogen Energy*, vol. 47, no. 33, pp. 14971–14982, 2022, doi: https://doi.org/10.1016/j.ijhydene.2022.02.239.

[27] X. Huang, Z. Li, Y. Jin, and W. Zhang, "Fair-AdaBoost: Extending AdaBoost method to achieve fair classification," *Expert Syst. Appl.*, vol. 202, p. 117240, 2022, doi: https://doi.org/10.1016/j.eswa.2022.117240.

[28] W. Wang and D. Sun, "The improved AdaBoost algorithms for imbalanced data classification," *Inf. Sci. (Ny).*, vol. 563, pp. 358–374, 2021, doi: https://doi.org/10.1016/j.ins.2021.03.042.

[29] S. Wan, X. Li, Y. Yin, and J. Hong, "Milling chatter detection by multi-feature fusion and Adaboost-SVM," *Mech. Syst. Signal Process.*, vol. 156, p. 107671, 2021, doi: https://doi.org/10.1016/j.ymssp.2021.107671.

[30] W. M. P.D. and Haryoko, "Optimization of parameter Support Vector Machine (SVM) using Genetic Algorithm to review Go-Jek's services," in *2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, 2019, pp. 301–304. doi: 10.1109/ICITISEE48480.2019.9003894.

[31] Sharazita Dyah Anggita and Ferian Fauzi Abdulloh, "Optimasi Algoritma Support Vector Machine berbasis PSO dan seleksi fitur Information Gain pada analisis sentimen," *J. Appl. Comput. Sci. Technol.*, vol. 4, no. 1, pp. 52–57, 2023, doi: 10.52158/jacost.v4i1.524.