Journal of Infrastructure, Policy and Development 2024, 8(9), 7101.
https://doi.org/10.24294/jipd.v8i9.7101

Article

Harmonizing sentiments: Analyzing user reviews of Spotify through sentiment analysis

Evaristus Didik Madyatmadja[1,*], Felix[1], I. Gusti Ketut Edrick[1], Joseph William Indarto[1], David Jumpa Malem Sembiring[2]

[1] Information Systems Department, School of Information Systems, Bina Nusantara University, Jakarta 11530, Indonesia
[2] Teknik Informatika, Institut Teknologi dan Bisnis Indonesia, Medan 20119, Indonesia
* **Corresponding author:** Evaristus Didik Madyatmadja, emadyatmadja@binus.edu

CITATION

Madyatmadja ED, Felix, Edrick IGK, et al. (2024). Harmonizing sentiments: Analyzing user reviews of Spotify through sentiment analysis. Journal of Infrastructure, Policy and Development. 8(9): 7101. https://doi.org/10.24294/jipd.v8i9.7101

ARTICLE INFO

Received: 12 June 2024
Accepted: 1 July 2024
Available online: 10 September 2024

Abstract: This research investigates the sentiment of user reviews on Spotify, with a particular focus on the Indonesian market, leveraging advanced sentiment analysis techniques. We employed three prominent classification models—Naïve Bayes, Support Vector Machine (SVM), and Random Forest—to analyze a dataset of 14,296 user reviews extracted from the Google Play Store and App Store. These findings reveal that the SVM model achieved the highest performance, with an $F1$-score of 0.875 and an accuracy of 0.874, outperforming Naïve Bayes and Random Forest, which scored accuracies of 0.857 and 0.856 respectively. These results highlight not only the significance of this research which offers valuable contributions to the broader academic discourse on digital marketing, sentiment analysis, and consumer behavior. Additionally, it also showcases the robustness and superior performance of SVM and Random Forest in various sentiment analysis contexts. This study not only provides valuable insights for Spotify's future development strategies but also contributes to the broader academic discourse on sentiment analysis and machine learning model performance in digital marketing. By highlighting the efficacy of specific models, this research underscores the importance of model selection in sentiment analysis, paving the way for more accurate and effective sentiment analysis applications in the music streaming industry.

Keywords: sentiment analysis; user reviews; Naïve Bayes; support vector machine; random forest; Spotify; Indonesia; music streaming

1. Introduction

In order to continue to gain an advantage over competitors, the use of sentiment analysis has become more widespread. One of the industries that can benefit from this method includes music streaming platforms, which have foreseen incredibly significant growth from 2018 (Muhamad, 2023a). By utilising this method, these platforms can benefit from assessing prevalent trends and sentiments that can prove valuable for their longevity and market value.

This research aims to analyze user sentiment on Spotify, specifically focusing on the Indonesian market. Leveraging sentiment analysis techniques can help to gain deeper insights into user perception and identify potential concerns arising from the recent price increase.

2. Background of the research

Music streaming platforms like Spotify have revolutionized music consumption, boasting exponential growth. Spotify, for instance, witnessed a threefold increase in premium subscribers between 2018 and 2023, reaching a staggering 220 million users by Q2 2023 (Muhamad, 2023a). This growth is mirrored by a rise in the global music industry's revenue, with streaming subscriptions contributing a significant 67% in 2022 (Annur, 2023). However, the Spotify price hike in July 2023 might influence user behavior and sentiment (Muhamad, 2023b).

In this dynamic market, understanding user sentiment is crucial for maintaining a competitive edge. Sentiment analysis of user reviews presents a valuable tool for gauging user satisfaction and pinpointing areas for improvement. This study will explore established classification models like Naïve Bayes, Random Forest, and Support Vector Machines, building upon the work of Madyatmadja et al. (2022) who achieved an accuracy of 94.2% using Support Vector Machines. The findings from this research will not only provide valuable insights for Spotify's future development but also contribute to the field of sentiment analysis by exploring its application in a specific cultural context.

2.1. Focus and significance of the research

This research delves into Spotify's remarkable market dominance in Indonesia through a multifaceted approach. Firstly, it dissects the interplay between market strategies and user preferences that fueled Spotify's explosive growth in premium subscribers between 2018 and 2023. This analysis aims to illuminate the key factors that propelled this expansion, offering valuable insights into successful digital marketing tactics and evolving user behavior within the music streaming landscape. Secondly, according to "Adjusting our Spotify premium prices (Muhamad, 2023b)", accessed on 12 May 2024, the research investigates the impact of Spotify's price hike for premium subscriptions implemented in July 2023 on the Indonesian market. By analyzing user sentiment and subscription choices, the study seeks to understand how these adjustments influence user perception and ultimately, their decisions to subscribe or discontinue services.

To gain a deeper understanding of user attitudes towards Spotify, a sentiment analysis of user reviews in Indonesia will be conducted. This analysis will specifically focus on user opinions regarding the recent price changes and overall service quality. By leveraging sentiment analysis techniques, the research aims to identify areas for improvement and gauge user satisfaction. Furthermore, the study will embark on a comparative analysis of established sentiment analysis models, including Naïve Bayes, Support Vector Machines (SVM), and Random Forest. This comparison will assess the accuracy and applicability of these models in interpreting user reviews within the music streaming industry. The goal is to determine the most effective approach for capturing user sentiment in this specific domain.

The significance of this research extends beyond immediate insights for Spotify. It offers valuable contributions to the broader academic discourse on digital marketing, sentiment analysis, and consumer behavior within the ever-evolving music industry. This research enriches scholarly dialogue by providing a nuanced understanding of user preferences and market dynamics in the context of music streaming services. Additionally, by exploring and comparing various sentiment analysis models, the study fosters innovation in data analytics methodologies for understanding consumer sentiment. This paves the way for the development of more advanced and accurate tools for gauging user opinions across diverse digital platforms. Ultimately, this research lays the groundwork for further studies in sentiment analysis, market trend analysis, and the impact of pricing strategies within the digitized music streaming market, contributing significantly to current understandings of the digital economy and consumer behavior.

2.2. ChatGPT (large language model)

ChatGPT, an advanced large language model (LLM) developed by OpenAI, stands as a pivotal innovation within the realm of sentiment analysis. Leveraging extensive pre-training, fine-tuning through instruction, and reinforcement learning from human feedback (RLHF), this sophisticated model has garnered acclaim for its remarkable adaptability and performance across diverse domains (Liu et al., 2023). Much akin to the meticulous examination conducted in sentiment analysis, ChatGPT showcases proficiency in comprehending, generating, and summarizing textual content across a nuanced spectrum, as observed across various works (Ogundare and Araya, 2023).

In the domain of Natural Language Processing (NLP), ChatGPT assumes a pivotal role in sentiment analysis, particularly in scrutinizing user feedback on platforms like Spotify. The model's exceptional performance has effectively bridged the gap between language generation and knowledge models, achieving feats that were previously reserved for specialized algorithms (Liu et al., 2023). The process of sentiment analysis entails a systematic progression through stages of data aggregation, text preprocessing, sentiment identification, classification, and presentation—a methodology familiar to researchers. Within this framework, researchers can opt for either Lexicon-based or machine learning-based approaches. While the former relies on predefined linguistic resources and quantitative assessment of lexical markers, ChatGPT's implementation under the machine learning paradigm leverages computational algorithms to discern intricate patterns within data (Ogundare and Araya, 2023).

This advanced approach involves the deployment of sophisticated algorithms trained on datasets annotated for sentiment, enabling nuanced predictions on novel textual inputs (Liu et al., 2023). Looking ahead, the research community anticipates exploring ChatGPT's potential beyond current applications. Academic interest in ChatGPT-related research is burgeoning, reflecting its promise in extending its utility to diverse domains such as education, history, mathematics, medicine, and physics. Moreover, investigations into ChatGPT's content detectability add another layer to its research potential (Zheng et al., 2023).

In summary, the integration of ChatGPT in sentiment analysis and NLP tasks presents a robust framework for analyzing user feedback on platforms like Spotify. The implications of this extend beyond conventional applications, offering profound opportunities for informed decision-making in diverse sectors, thereby shaping sentiments in this digitally interconnected landscape (Zhang, 2023).

2.3. Sentiment analysis

Sentiment analysis, also termed opinion mining, constitutes a computational approach employed to extract, categorize, and scrutinize emotions, opinions, and attitudes embedded within textual corpora. This analytical process endeavors to unravel the nuanced array of sentiments encapsulated within textual constructs, traversing the continuum from positivity to negativity, with a neutral standpoint occupying an intermediary position (Aqlan et al., 2019). Situated within the domain of Natural Language Processing (NLP), sentiment analysis emerges as an indispensable tool, tasked with the formidable challenge of deciphering individuals' sentiments vis-à-vis diverse entities such as products, services, or societal concerns, disseminated across the digital landscape (Yadav, 2023).

Diverse methodologies underpin sentiment analysis, encompassing lexicon-based approaches and machine learning-driven algorithms, each serving to discern intricate patterns within textual data. Lexicon-based strategies lean on predefined linguistic resources to gauge the prevalence of positive and negative lexical cues within textual inputs, thus delineating overarching sentiment (Drus and Khalid, 2019). Conversely, machine learning-driven paradigms harness computational algorithms to predict nuanced sentiment across novel textual inputs (Tripathy and Rath, 2017). The ramifications of sentiment analysis are manifold and profound, permeating diverse spheres including political discourse, healthcare, prognostication, and security. It functions as a potent instrument for gauging public sentiment, prognosticating behavioral trajectories, and unearthing latent issues warranting deeper scrutiny (Usha and Dharmanna, 2021).

Looking ahead, sentiment analysis emerges as a fertile ground for continued exploration and innovation. Recent strides in deep learning-based sentiment analysis, as evidenced by Yuan et al. (2020), promise heightened accuracy and efficiency in sentiment classification endeavors. Moreover, the deployment of sentiment analysis extends into varied domains, such as the classification of forest fires in Indonesia, as elucidated by Irawanto et al. (2023), indicating its expansive utility. Furthermore, the fusion of sentiment analysis with sophisticated language models, as elucidated in the study by Kumawat et al. (2021), unlocks avenues for leveraging advanced language models to facilitate more nuanced sentiment prediction and analysis. In summation, sentiment analysis, with its multifaceted applications and potential for integration with cutting-edge technologies, emerges as an intellectually stimulating and promising frontier for further research and exploration (Aqlan et al., 2019).

3. Related works

Previous research has extensively utilized sentiment analysis to classify documents, especially with machine learning approaches. Studies have predominantly focused on popular algorithms such as Support Vector Machine (SVM), Naïve Bayes, and Random Forest classifiers. This section explores various studies that have applied these algorithms to different datasets, evaluating their effectiveness in sentiment analysis and highlighting significant conclusions.

The application of machine learning techniques in sentiment analysis is highly debated and contested. Madyatmadja et al. (2023) argue that Random Forest and

Support Vector Machine (SVM) provide higher accuracy rates compared to other algorithms in classifying crowdsourced citizen complaints, demonstrating their robustness in handling complex and diverse datasets. This view is supported by Dake and Gyimah (2023), who also found that SVM and Random Forest achieved the highest accuracy rates in evaluating qualitative student responses, reaffirming the reliability of these algorithms in educational data contexts. However, Madyatmadja et al. (2023) did not delve into factors influencing the effectiveness of these algorithms, such as data preprocessing and feature selection.

On one hand, Mamun et al. (2022) contend that ensemble methods, which combine multiple classifiers, offer superior performance over individual models. Their results indicated that an ensemble approach incorporating Random Forest and SVM yielded the highest accuracy, highlighting the benefits of leveraging multiple algorithms to improve sentiment classification outcomes. This perspective is echoed by Varshney et al. (2020), who found that the ensemble approach significantly improved classification accuracy, with the ensemble model outperforming individual classifiers. However, Mamun et al. (2022) lack a discussion on the specific contributions of each classifier within the ensemble.

On the other hand, Dr. Arivoli and Sonali (2021) emphasize the importance of feature selection in improving classifier performance. They argue that incorporating feature selection significantly enhances the performance of SVM, making it the most accurate model in their study on sentiment analysis using feature selection and semantic analysis. This emphasis on feature selection is crucial, as it is often overlooked in other studies, a point that is backed up by Indulkar and Patil (2021). They conducted a comparative study of machine learning algorithms for Twitter sentiment analysis, finding that SVM and Random Forest were the top performers. However, they noted that the influence of different preprocessing techniques on the algorithms' performance is a critical area that needs further exploration.

Furthermore, Aufar et al. (2020) highlight the robustness of Random Forest in handling large and varied datasets typical of social media platforms. Their investigation into sentiment analysis on YouTube social media using Decision Tree and Random Forest algorithms indicated that Random Forest outperformed Decision Tree in terms of accuracy. However, their study lacks a critical evaluation of how data characteristics affect model performance, a gap that Velmurugan et al. (2022) also identify in their examination of customer reviews using various classification algorithms, including Naïve Bayes, SVM, Random Forest, and Decision Tree.

Meanwhile, Madyatmadja et al. (2022) performed sentiment analysis on user reviews of mutual fund applications using various machine learning techniques. They found that SVM and Random Forest outperformed other classifiers, achieving the highest accuracy rates. This work underscores the effectiveness of these algorithms in processing financial review data, showcasing their potential in sentiment analysis within the financial sector. Nevertheless, the study did not address how the characteristics of the dataset might impact the performance of these algorithms.

Finally, Singh et al. (2020) provided a comprehensive review and comparative analysis of sentiment analysis techniques over social media data. They identified SVM and Random Forest as consistently high-performing algorithms across multiple studies, further validating their effectiveness in sentiment classification tasks. While comprehensive, their review does not critically assess the influence of dataset characteristics on model performance, a critical aspect that needs more attention in future research.

The primary distinction between the previous efforts and this research is the comprehensive comparison of multiple machine learning algorithms: Naïve Bayes, Support Vector Machines, and Random Forests. Unlike prior studies, this research investigates the influence of dataset characteristics on the effectiveness of these classifiers in sentiment analysis of Spotify user reviews. By evaluating the efficacy of several algorithms and exploring the underlying reasons driving their performance, our study provides a deeper understanding of sentiment analysis approaches and highlights critical factors affecting model quality, such as data preprocessing, feature selection, and parameter tuning. This comprehensive method aims to advance the field by offering nuanced insights into the application of machine learning techniques for sentiment analysis in the context of Spotify user reviews.

4. Material and methods

The sentiment analysis for Spotify is preceded with web scraping to collect data from the Google Play Store and App Store using the google-play-scraper and app-store-scraper library in a python jupyter notebook environment. This data is then processed through text preprocessing and feature extraction. To further refine the data, the following six steps are taken: normalization, case folding, stop word removal, tokenization, stemming, and labeling. Finally, the data is split into training and testing subsets for sentiment analysis classification, followed by evaluation, comparison, and visualization to gain a comprehensive understanding of the results.

4.1. Data collection

As illustrated in **Figure 1**, the google-play-scraper and app-store-scraper library is used to gather data from the Google Play Store and App Store within a timeframe of 3 November 2023, up to 23 March 2024. The collected data is imported into the jupyter notebook environment and exported into CSV format for further analysis. The dataset contains various columns with different attributes and functionalities. Then, the data is further streamlined by retaining only the columns relevant to sentiment analysis. The content, ratings, and label columns are defined or created later during data processing. This approach provides a clean dataset for subsequent analysis.

Figure 1. Data collection method.

4.2. Text preprocessing

In the text preprocessing stage, depicted in **Figure 2** below, the data collected from the Google Play Store and App Store is cleaned and standardized before continuing to the sentiment analysis.

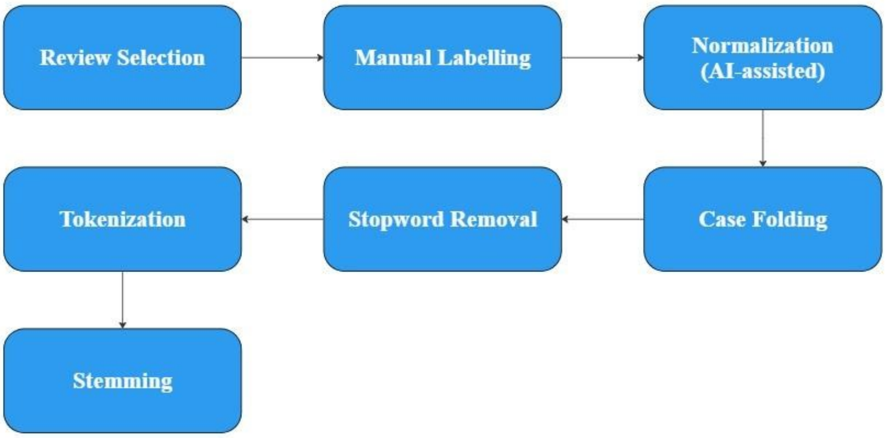


Figure 2. Text preprocessing method.

The first step is normalization, where slang, abbreviations, and non-standard spellings are corrected based on KBBI (Kamus Besar Bahasa Indonesia). To enhance this process, we use ChatGPT, a large language model (LLM) which are illustrated on **Figures 3–5**, to help the mapping of non-standard words to their standard forms. The normalization mapping includes common corrections like "Fiitur" to "Fitur," "sportify" to "Spotify," and "lelet" to "lambat," among others.

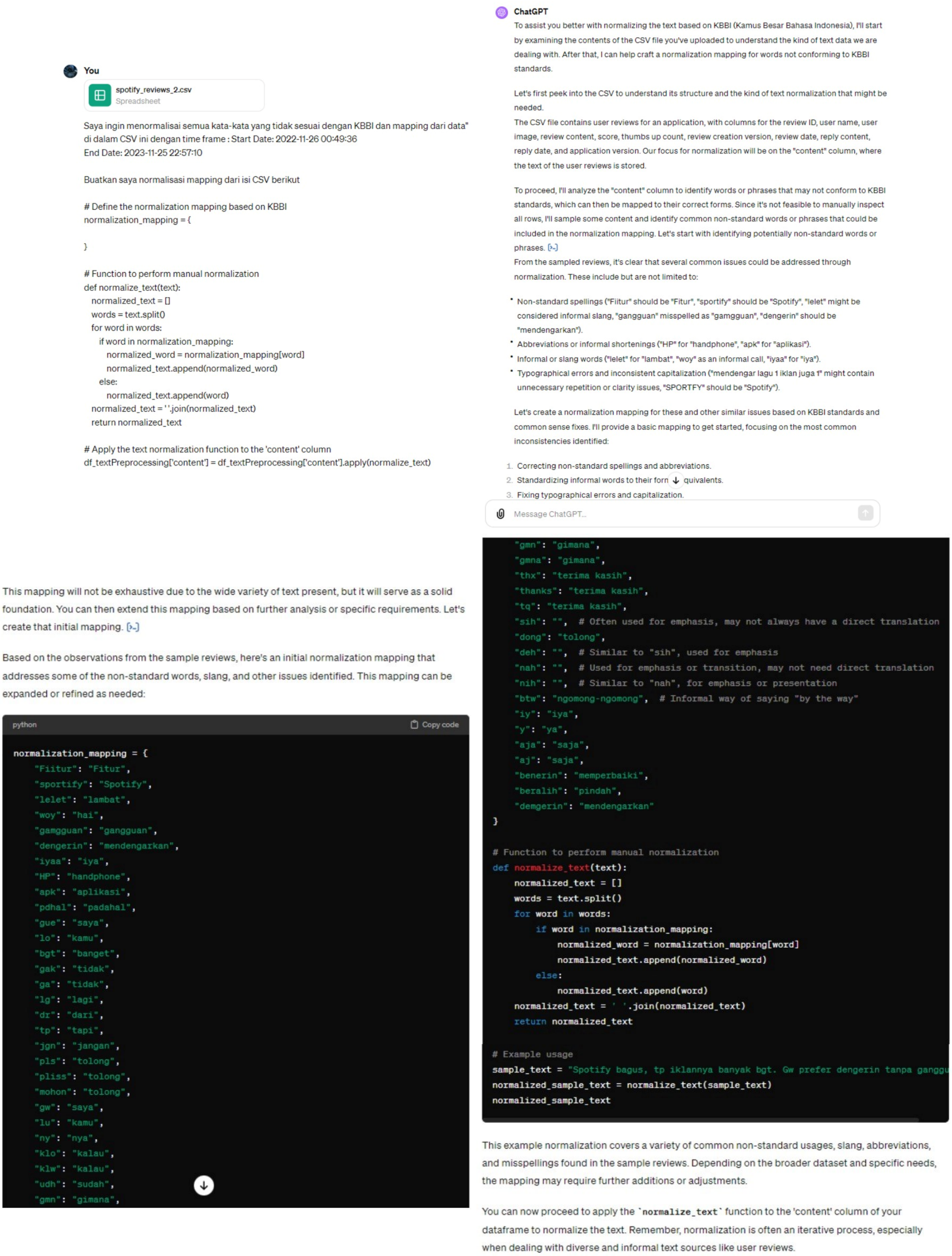
The next step is to perform case folding by converting all text to lowercase using the NLTK library. This is done in order to ensure consistency across the dataset. Following that, stop word removal is conducted to eliminate common conjunctions and filler words that do not contribute to sentiment analysis. The Sastrawi library is used to remove these words, along with additional manual adjustments based on specific needs.

Tokenization is another critical step, where text is split into individual words for easier analysis. This step is completed using NLTK's tokenization methods, allowing for more detailed examination of the data. Lastly, stemming reduces words to their root forms to create a clearer understanding of the data. This step is achieved with the Sastrawi stemmer factory, providing efficient stemming for Bahasa Indonesia.

Given the importance of consistent opinions and novelty, only reviews with 1-, 2-, 4-, or 5-star ratings are retained for this study. Reviews with 3-star ratings are considered neutral and are removed from the dataset. The ratings are further labeled accordingly based on their values: 1- and 2-star reviews are labeled as "negative", while 4- and 5-star reviews are labeled as "positive". This labeling is done to ensure a clear categorization for sentiment analysis.

This structured approach ensures that the text preprocessing process is robust and tailored to the needs of this sentiment analysis study, with a timeframe from 6 October 2023 to 23 March 2024. Below is a tailored explanation on how researchers utilize large language models to assist in the research methodology.

To get a clear understanding of the text data in a CSV file, we can start by asking ChatGPT to create a normalization mapping based on Kamus Besar Bahasa Indonesia (KBBI) as shown in **Figure 3**. This helps correct non-standard words, slang, and abbreviations for a consistent dataset.



You

spotify_reviews_2.csv
Spreadsheet

Saya ingin menormalisai semua kata-kata yang tidak sesuai dengan KBBI dan mapping dari data" di dalam CSV ini dengan time frame : Start Date: 2022-11-26 00:49:36
End Date: 2023-11-25 22:57:10

Buatkan saya normalisasi mapping dari isi CSV berikut

```
# Define the normalization mapping based on KBBI
normalization_mapping = {

}

# Function to perform manual normalization
def normalize_text(text):
  normalized_text = []
  words = text.split()
  for word in words:
    if word in normalization_mapping:
      normalized_word = normalization_mapping[word]
      normalized_text.append(normalized_word)
    else:
      normalized_text.append(word)
  normalized_text = ' '.join(normalized_text)
  return normalized_text

# Apply the text normalization function to the 'content' column
df_textPreprocessing['content'] = df_textPreprocessing['content'].apply(normalize_text)
```

ChatGPT

To assist you better with normalizing the text based on KBBI (Kamus Besar Bahasa Indonesia), I'll start by examining the contents of the CSV file you've uploaded to understand the kind of text data we are dealing with. After that, I can help craft a normalization mapping for words not conforming to KBBI standards.

Let's first peek into the CSV to understand its structure and the kind of text normalization that might be needed.

The CSV file contains user reviews for an application, with columns for the review ID, user name, user image, review content, score, thumbs up count, review creation version, review date, reply content, reply date, and application version. Our focus for normalization will be on the "content" column, where the text of the user reviews is stored.

To proceed, I'll analyze the "content" column to identify words or phrases that may not conform to KBBI standards, which can then be mapped to their correct forms. Since it's not feasible to manually inspect all rows, I'll sample some content and identify common non-standard words or phrases that could be included in the normalization mapping. Let's start with identifying potentially non-standard words or phrases. [>-]

From the sampled reviews, it's clear that several common issues could be addressed through normalization. These include but are not limited to:

- Non-standard spellings ("Fiitur" should be "Fitur", "sportify" should be "Spotify", "lelet" might be considered informal slang, "gangguan" misspelled as "gamgguan", "dengerin" should be "mendengarkan").
- Abbreviations or informal shortenings ("HP" for "handphone", "apk" for "aplikasi").
- Informal or slang words ("lelet" for "lambat", "woy" as an informal call, "iyaa" for "iya").
- Typographical errors and inconsistent capitalization ("mendengar lagu 1 iklan juga 1" might contain unnecessary repetition or clarity issues, "SPORTFY" should be "Spotify").

Let's create a normalization mapping for these and other similar issues based on KBBI standards and common sense fixes. I'll provide a basic mapping to get started, focusing on the most common inconsistencies identified:

1. Correcting non-standard spellings and abbreviations.
2. Standardizing informal words to their forn quivalents.
3. Fixing typographical errors and capitalization.

Message ChatGPT...

This mapping will not be exhaustive due to the wide variety of text present, but it will serve as a solid foundation. You can then extend this mapping based on further analysis or specific requirements. Let's create that initial mapping. [>-]

Based on the observations from the sample reviews, here's an initial normalization mapping that addresses some of the non-standard words, slang, and other issues identified. This mapping can be expanded or refined as needed:

python Copy code

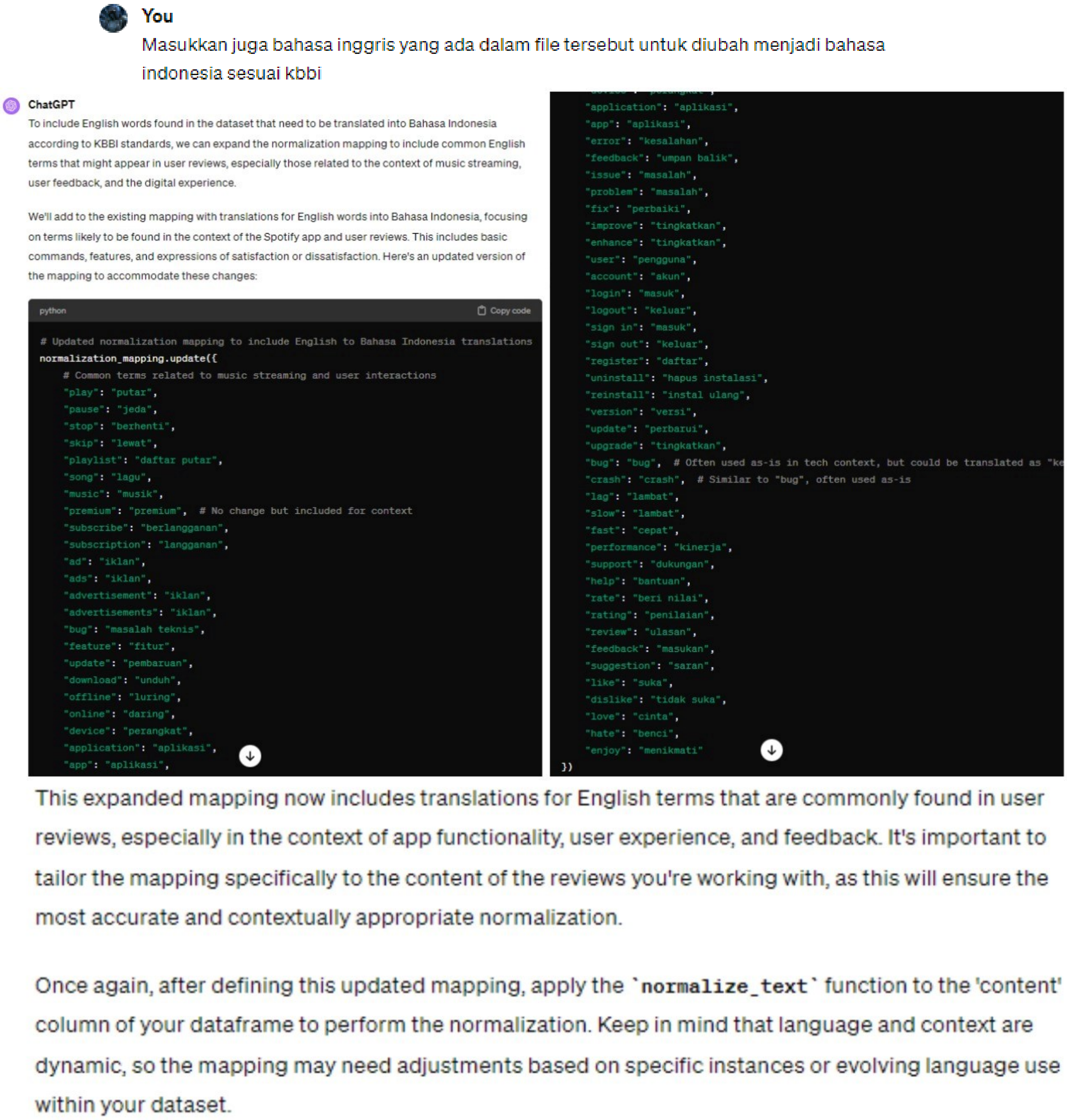
```python
normalization_mapping = {
    "Fiitur": "Fitur",
    "sportify": "Spotify",
    "lelet": "lambat",
    "woy": "hai",
    "gamgguan": "gangguan",
    "dengerin": "mendengarkan",
    "iyaa": "iya",
    "HP": "handphone",
    "apk": "aplikasi",
    "pdhal": "padahal",
    "gue": "saya",
    "lo": "kamu",
    "bgt": "banget",
    "gak": "tidak",
    "ga": "tidak",
    "lg": "lagi",
    "dr": "dari",
    "tp": "tapi",
    "jgn": "jangan",
    "pls": "tolong",
    "pliss": "tolong",
    "mohon": "tolong",
    "gw": "saya",
    "lu": "kamu",
    "ny": "nya",
    "klo": "kalau",
    "klw": "kalau",
    "udh": "sudah",
    "gmn": "gimana",
```

```python
    "gmn": "gimana",
    "gmna": "gimana",
    "thx": "terima kasih",
    "thanks": "terima kasih",
    "tq": "terima kasih",
    "sih": "",  # Often used for emphasis, may not always have a direct translation
    "dong": "tolong",
    "deh": "",  # Similar to "sih", used for emphasis
    "nah": "",  # Used for emphasis or transition, may not need direct translation
    "nih": "",  # Similar to "nah", for emphasis or presentation
    "btw": "ngomong-ngomong",  # Informal way of saying "by the way"
    "iy": "iya",
    "y": "ya",
    "aja": "saja",
    "aj": "saja",
    "benerin": "memperbaiki",
    "beralih": "pindah",
    "demgerin": "mendengarkan"
}

# Function to perform manual normalization
def normalize_text(text):
    normalized_text = []
    words = text.split()
    for word in words:
        if word in normalization_mapping:
            normalized_word = normalization_mapping[word]
            normalized_text.append(normalized_word)
        else:
            normalized_text.append(word)
    normalized_text = ' '.join(normalized_text)
    return normalized_text

# Example usage
sample_text = "Spotify bagus, tp iklannya banyak bgt. Gw prefer dengerin tanpa ganggu
normalized_sample_text = normalize_text(sample_text)
normalized_sample_text
```

This example normalization covers a variety of common non-standard usages, slang, abbreviations, and misspellings found in the sample reviews. Depending on the broader dataset and specific needs, the mapping may require further additions or adjustments.

You can now proceed to apply the `normalize_text` function to the 'content' column of your dataframe to normalize the text. Remember, normalization is often an iterative process, especially when dealing with diverse and informal text sources like user reviews.

Figure 3. Researcher request and GPT responses 1.



You

Masukkan juga bahasa inggris yang ada dalam file tersebut untuk diubah menjadi bahasa indonesia sesuai kbbi

ChatGPT

To include English words found in the dataset that need to be translated into Bahasa Indonesia according to KBBI standards, we can expand the normalization mapping to include common English terms that might appear in user reviews, especially those related to the context of music streaming, user feedback, and the digital experience.

We'll add to the existing mapping with translations for English words into Bahasa Indonesia, focusing on terms likely to be found in the context of the Spotify app and user reviews. This includes basic commands, features, and expressions of satisfaction or dissatisfaction. Here's an updated version of the mapping to accommodate these changes:

python   Copy code

```python
# Updated normalization mapping to include English to Bahasa Indonesia translations
normalization_mapping.update({
    # Common terms related to music streaming and user interactions
    "play": "putar",
    "pause": "jeda",
    "stop": "berhenti",
    "skip": "lewat",
    "playlist": "daftar putar",
    "song": "lagu",
    "music": "musik",
    "premium": "premium",  # No change but included for context
    "subscribe": "berlangganan",
    "subscription": "langganan",
    "ad": "iklan",
    "ads": "iklan",
    "advertisement": "iklan",
    "advertisements": "iklan",
    "bug": "masalah teknis",
    "feature": "fitur",
    "update": "pembaruan",
    "download": "unduh",
    "offline": "luring",
    "online": "daring",
    "device": "perangkat",
    "application": "aplikasi",
    "app": "aplikasi",
```

```python
    "application": "aplikasi",
    "app": "aplikasi",
    "error": "kesalahan",
    "feedback": "umpan balik",
    "issue": "masalah",
    "problem": "masalah",
    "fix": "perbaiki",
    "improve": "tingkatkan",
    "enhance": "tingkatkan",
    "user": "pengguna",
    "account": "akun",
    "login": "masuk",
    "logout": "keluar",
    "sign in": "masuk",
    "sign out": "keluar",
    "register": "daftar",
    "uninstall": "hapus instalasi",
    "reinstall": "instal ulang",
    "version": "versi",
    "update": "perbarui",
    "upgrade": "tingkatkan",
    "bug": "bug",  # Often used as-is in tech context, but could be translated as "ke
    "crash": "crash",  # Similar to "bug", often used as-is
    "lag": "lambat",
    "slow": "lambat",
    "fast": "cepat",
    "performance": "kinerja",
    "support": "dukungan",
    "help": "bantuan",
    "rate": "beri nilai",
    "rating": "penilaian",
    "review": "ulasan",
    "feedback": "masukan",
    "suggestion": "saran",
    "like": "suka",
    "dislike": "tidak suka",
    "love": "cinta",
    "hate": "benci",
    "enjoy": "menikmati"
})
```

This expanded mapping now includes translations for English terms that are commonly found in user reviews, especially in the context of app functionality, user experience, and feedback. It's important to tailor the mapping specifically to the content of the reviews you're working with, as this will ensure the most accurate and contextually appropriate normalization.

Once again, after defining this updated mapping, apply the `normalize_text` function to the 'content' column of your dataframe to perform the normalization. Keep in mind that language and context are dynamic, so the mapping may need adjustments based on specific instances or evolving language use within your dataset.

Figure 4. Researcher request and GPT responses 2.

Once the prompt has returned a reply, the next step is examining the data to identify non-standard language for normalization. The guidance suggests focusing on the content column since that's where most text data is stored. Then, words or phrases that need correction are revised and mapped into proper forms by using KBBI as a reference.

ChatGPT provided a detailed normalization mapping with common examples of non-standard words and their corrections. This prompt mapping addresses various issues, like slang, abbreviations, and typographical errors, further showing how words should be adjusted to comply with KBBI standards.

An example function, normalize text, is created to process the text data. It splits the text into individual words, checks them against the normalization mapping, and replaces non-standard words with their standard equivalents. If a word does not require correction, it is kept as is.

After applying the normalization function, the resulting text reflects the expected changes. A sample text demonstrates how common slang, and abbreviations are corrected, ensuring consistency. This process lays the groundwork for text preprocessing, feature extraction, and sentiment analysis, providing a cleaner dataset that's ready for further analysis and modeling.

In **Figure 4**, we prompted the normalization mapping to include translations for English terms that need to be converted into Indonesian according to Kamus Besar Bahasa Indonesia (KBBI). This approach is crucial for ensuring that common English words in user reviews, especially those related to the Spotify app, are correctly translated to maintain consistency.

The mapping provided by ChatGPT reflects this expanded scope. It incorporates common English terms found in user reviews, particularly in the context of music streaming, user feedback, and digital services. For example, words related to music interaction, like "play," "pause," or "skip," are translated into their appropriate Bahasa Indonesia terms, such as "putar," "jeda," and "lewat." The mapping also addresses other common terms, including "playlist" (translated to "daftar putar") and "premium" (which remains unchanged but is included for context).

After defining this expanded mapping, the next step is to apply the normalize-text function to the "content" column of the dataset. This function takes each word, checks it against the normalization mapping, and replaces non-standard English words with their corresponding Indonesian translations. The mapping ensures a thorough correction of English terms, resulting in a more consistent and comprehensible dataset.

It's important to note that the mapping might require further adjustments to accommodate language variations and evolving contexts. This updated normalization provides a solid foundation for further text preprocessing, feature extraction, and sentiment analysis, facilitating a smooth analysis of user reviews.

In **Figure 5**, we focus on optimizing the text preprocessing and feature extraction steps for handling Bahasa Indonesia, combining the best elements from various libraries. Given the need for efficient processing in sentiment analysis, this method ensures high-quality results by selecting the most suitable libraries and incorporating user-defined customizations.

First, stopword removal is addressed. NLTK's list of stopwords is used, designed to remove common words that do not contribute to sentiment analysis. If there are specific stopwords not covered by NLTK, the list can be adjusted to suit domain-specific requirements based on the CSV file. This flexibility allows you to manage stopword removal in a way that best fits your data.

Next is tokenization. Here, NLTK's tokenizer is chosen for its versatility and capability to work across different languages. This step divides the text into smaller components, making it easier to analyze and process. After tokenization, normalization is applied, using a mapping created earlier to convert slang, abbreviations, and non-standard words to their proper forms before further processing.

Stemming uses the Sastrawi library, which is tailored for Indonesian. This step reduces words to their root forms, aiding consistency and ensuring that different variations of the same word are treated equally. Sastrawi's specialization in Indonesian stemming makes it an ideal choice for this stage.

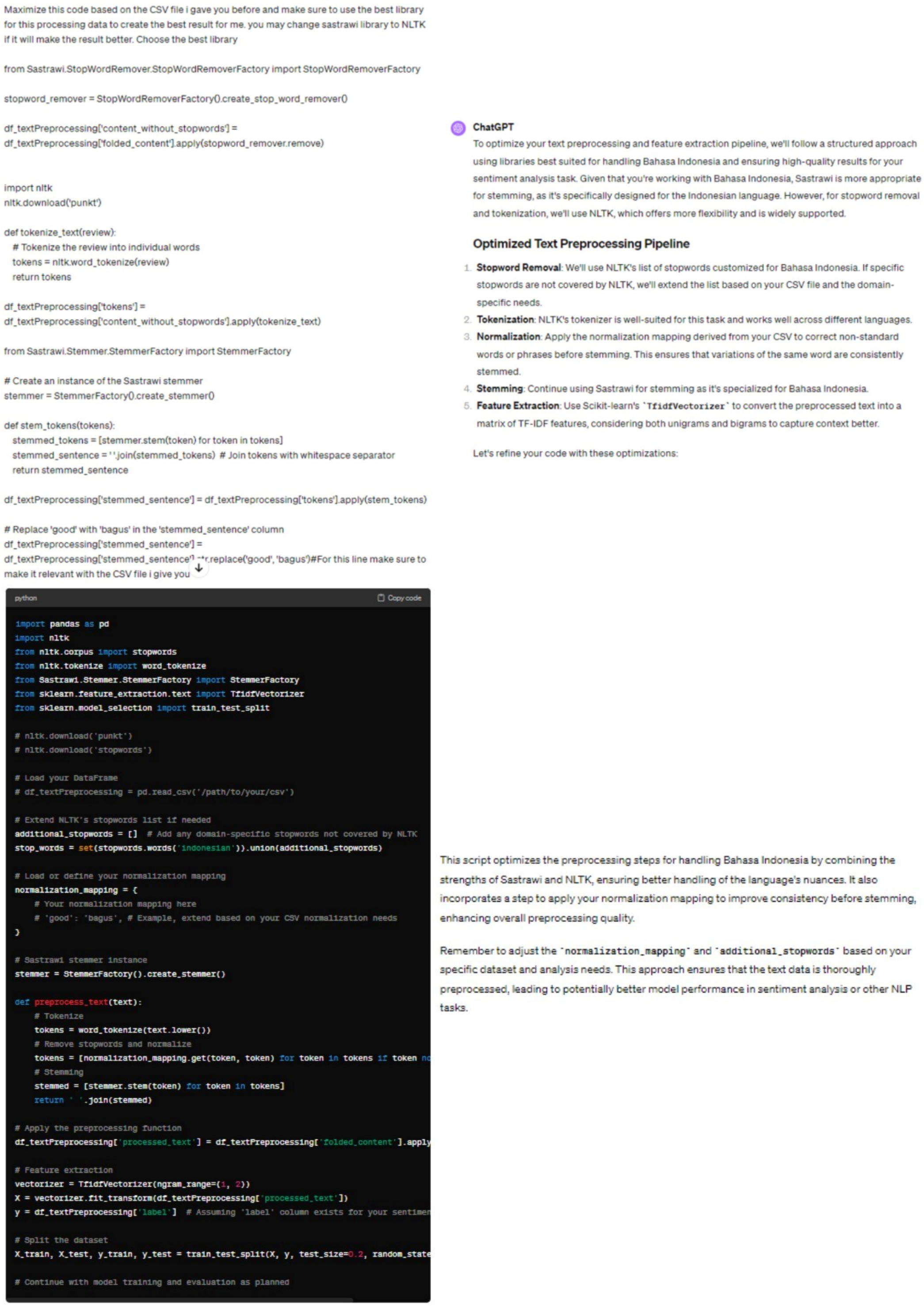


Figure 5. Researcher request and GPT responses 3.

In **Figure 5**, we focus on optimizing the text preprocessing and feature extraction steps for handling Bahasa Indonesia, combining the best elements from various libraries. Given the need for efficient processing in sentiment analysis, this method ensures high-quality results by selecting the most suitable libraries and incorporating user-defined customizations.

First, stopword removal is addressed. NLTK's list of stopwords is used, designed to remove common words that do not contribute to sentiment analysis. If there are specific stopwords not covered by NLTK, the list can be adjusted to suit domain-specific requirements based on the CSV file. This flexibility allows you to manage stopword removal in a way that best fits your data.

Next is tokenization. Here, NLTK's tokenizer is chosen for its versatility and capability to work across different languages. This step divides the text into smaller components, making it easier to analyze and process. After tokenization, normalization is applied, using a mapping created earlier to convert slang, abbreviations, and non-standard words to their proper forms before further processing.

Stemming uses the Sastrawi library, which is tailored for Indonesian. This step reduces words to their root forms, aiding consistency and ensuring that different variations of the same word are treated equally. Sastrawi's specialization in Indonesian stemming makes it an ideal choice for this stage.

Feature extraction focuses on converting preprocessed text into a numerical format suitable for analysis and modeling. Scikit-learn's TfidfVectorizer is used, allowing text to be transformed into a matrix of TF-IDF features, considering both unigrams and bigrams. This approach captures context and helps in identifying important terms within the data.

To maintain flexibility and ensure optimal results, the script allows for adjustments to the normalization mapping and additional stopwords based on the content of the CSV file. This adaptability ensures that the data is thoroughly preprocessed, leading to potentially better model performance in sentiment analysis or other NLP tasks.

With these steps, the preprocessing pipeline becomes robust, efficiently handling the nuances of Indonesian and providing a solid foundation for sentiment analysis and feature extraction.

4.3. Features extraction

After text preprocessing, the next step is to extract features, which is illustrated in **Figure 6**, that will be used for analysis and modeling. This process transforms the text into a numerical format that machine learning models can work with. In this study, there are 14,296 records in the text preprocessing data, which is then converted into a numerical format.

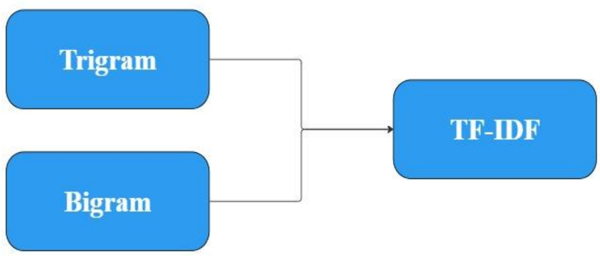


Figure 6. Feature extraction method.

One of the primary methods utilised is TF-IDF (Term Frequency-Inverse Document Frequency). This approach assigns weight to each word based on its frequency within a document and across the entire dataset. Words that appear frequently within a specific document but are less common across the dataset receive higher weights, while words that are widespread receive lower weights. This method highlights key terms that are significant for distinguishing documents.

To maintain context and understand word sequences, *n*-grams are used. *N*-grams capture pairs or groups of consecutive words, such as bigrams (two-word sequences) and trigrams (three-word sequences). This method helps preserve the word order, which is crucial for analyzing complex patterns in the text. Using these feature extraction techniques, this creates a solid foundation for sentiment analysis and machine learning-based modeling.

4.4. Sentiment analysis classification

Sentiment analysis classification, as depicted in **Figure 7**, involves categorizing reviews into positive or negative sentiments. The data is divided into two subsets—training and testing—using an 80:20 ratio, which helps ensure a balanced approach for training and testing the models. The training data is used to build the models, while the testing data is used to evaluate their performance.

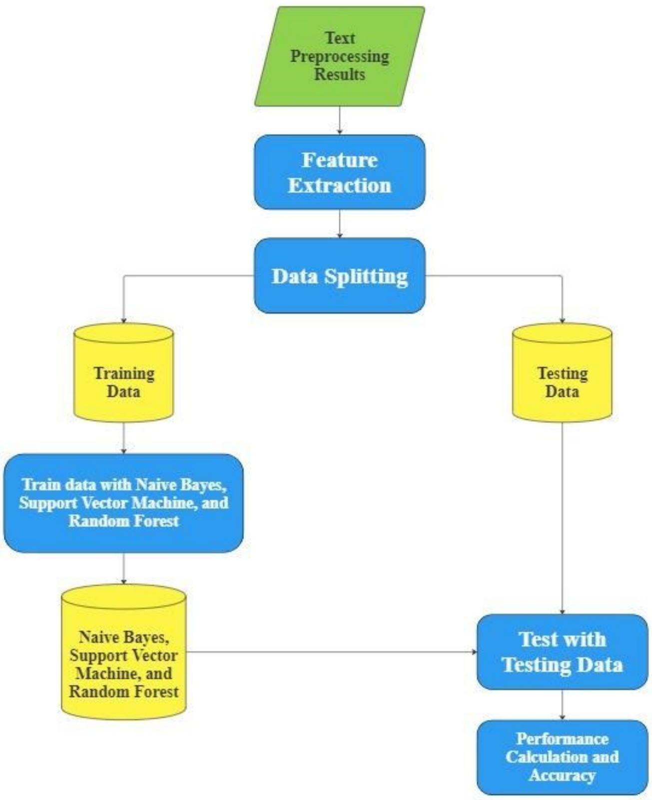


Figure 7. Sentiment analysis classification method.

The sentiment analysis classification methods used in this study include Naïve Bayes, Support Vector Machine (SVM), and Random Forest. Naïve Bayes is a probabilistic classifier that predicts sentiment based on word frequencies. Support

Vector Machine (SVM) is a supervised learning model designed for high accuracy and robustness. Random Forest is an ensemble method that combines multiple decision trees to improve classification accuracy. By using these three models, the data is categorised into positive or negative groups, allowing for effective analysis.

4.5. Evaluation and comparison

To evaluate the performance of these classification models, we use a variety of metrics that measure their accuracy and effectiveness. The confusion matrix is one such metric, providing a detailed view of classification results. It calculates true positives, true negatives, false positives, and false negatives to give a complete picture of how well the models are working.

Precision, recall, and $F1$ score are also perused to ensure a comprehensive assessment of each classification method. Precision measures the proportion of correct positive predictions, recall calculates the proportion of actual positives correctly identified, and $F1$ score is the harmonic mean of precision and recall. Additionally, overall accuracy offers a general measure of the classification models' success in categorizing reviews into positive or negative sentiments.

4.6. Visualization

For visualization, bar charts are used to compare precision, recall, $F1$ score, and accuracy across different classification models. This provides a clear visual representation of the performance of each model. Word clouds are used to visualize the most frequent words in positive and negative reviews, giving us insights into common themes and patterns. These visualization techniques help us evaluate and compare the classification models used in this study.

5. Results and discussion

This research collected 14,296 user reviews extracted from Google Play Store and Apps Store. Said data has undergone text processing stages to ensure its cleanliness and suitability for sentiment analysis purposes. As illustrated in **Table 1**, the data was subsequently split with an 80:20 ratio, resulting in 11,436 training data samples and 2860 testing data samples. The research reveals that over 1900 positive testing exhibits positive sentiments, focusing on user experiences with the application, while the remaining contain negative reviews that highlight subscription issues and limitations in free features.

Table 1. Numbers of data split for each label.

Label	Data numbers
Training	11,436
Testing	2860

5.1. Classification model

5.1.1. Naïve bayes

The multinomial Naive Bayes (NB) model is utilized for classification tasks, accessed through scikit-learn package and NLTK function, and was employed for data classification based on keywords. An examination of the Naive Bayes method using the confusion matrix (**Table 2**), reveals specific performance metrics: 1808 true positives (TP), 642 true negatives (TN), 142 false positives (FP), and 268 false negatives (FN). Evaluation metrics in **Table 3** demonstrate an average precision of 84%, an average recall of 82%, an *f*1-score of 83%, and an overall accuracy of 86% for the Naive Bayes model. These computations offer a comprehensive insight into the classification performance of the Naive Bayes model.

Table 2. Confusion matrix of Naïve Bayes model.

	Positive	**Negative**
Predicted positive	1808	142
Predicted negative	268	642

Table 3. Classification report of Naïve Bayes model.

	Precision	**Recall**	***F*1-score**	**Support**
Negative	0.82	0.71	0.76	910
Positive	0.87	0.93	0.90	1950
Accuracy			0.86	2860
Macro avg	0.84	0.82	0.83	2860
Weighted avg	0.85	0.86	0.85	2860

5.1.2. Support vector machine

In developing the classification model, the Support Vector Machine (SVM) technique was implemented, specifically employing the Support Vector Classifier (SVC) with a linear kernel through the scikit-learn library in python. After constructing the model, predictions were generated on the testing dataset. The evaluation of the SVM model, outlined in **Table 4**, reveals specific performance metrics: 1744 true positives (TP), 239 false positives (FP), 755 true negatives (TN), and 155 data false negatives (FN) in the confusion metrics. Referring to **Table 5**, it detailed the evaluation metrics showing an average precision 85%, average recall 86%, *f*1-score 86%, and the accuracy of SVM model 87%. It can be concluded that the accuracy of the SVM model demonstrated a strong performance and surpassed the other models' accuracy used in this research.

Table 4. Confusion matrix of SVM model.

	Positive	**Negative**
Predicted positive	1744	206
Predicted negative	155	755

Table 5. Classification report of SVM model.

	Precision	Recall	*F*1-score	Support
Negative	0.79	0.83	0.81	910
Positive	0.92	0.89	0.91	1950
Accuracy			0.87	2860
Macro avg	0.85	0.86	0.86	2860
Weighted avg	0.88	0.87	0.87	2860

5.1.3. Random Forest

The Random Forest (RF) classification method operates by employing numerous decision trees from random subsets of data. Each tree contributes to the final classification by voting for a particular class. By examining the performance of Random Forest model through provided confusion matrix, the following outcomes were observed: 1711 true positive (TP), 239 false positive (FP), 737 true negative (TN), and 173 false negative (FN), detailed in **Table 6**. In this research, the Random Forest model achieved an accuracy rate of 86%. The calculation of the evaluation results can be seen in **Table 7**.

Table 6. Confusion matrix of Random Forest model.

	Positive	Negative
Predicted positive	1711	239
Predicted negative	173	737

Table 7. Classification report of Random Forest model.

	Precision	Recall	*F*1-score	Support
Negative	0.76	0.81	0.78	910
Positive	0.91	0.88	0.89	1950
Accuracy			0.86	2860
Macro avg	0.83	0.84	0.84	2860
Weighted avg	0.86	0.86	0.86	2860

5.2. Sentiment analysis results

After assessment and visualisation of the dataset of Spotify user reviews obtained from Google Play Store and Apps Store, this research study is able to identify the main reason(s) behind their positive or negative sentiments to the Spotify application. In the preprocessing step, the dataset is cleaned and relevant words for analysis and visualization purposes are gathered. Then, the frequency of each word is shown in a bar chart generated using matplotlib. The bar chart visualizes the most common word that users use in their review, either positive or negative reviews.

The chart shown in **Figure 8** displays the top 10 words that users use to express their positive sentiment in the application review. These words include ‘bagus’, ‘lagu’, ‘aplikasi’, ‘iklan’, ‘musik’, ‘spotify’, ‘banget’, ‘dengar’, ‘suka’, and ‘lirik’. The most popular positive word is ‘bagus’ which translates to ‘good’ with over 2500 frequencies. The word ‘bagus’ is frequently seen in the positive review on the application corresponding with the meaning. This word gives a very positive sentiment to the Spotify application which means the users are very pleased in using Spotify as their daily music streaming application.

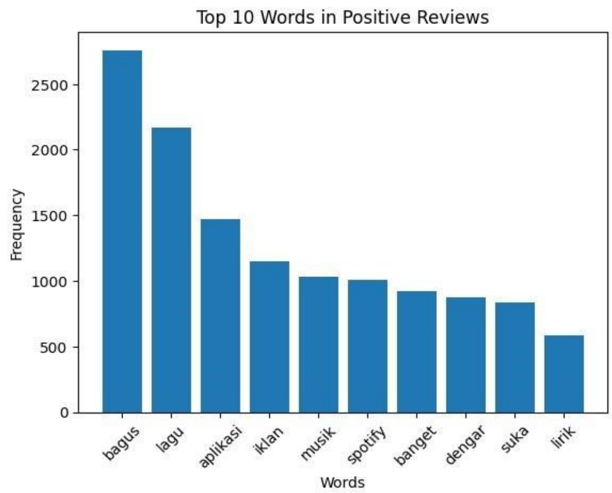


Figure 8. Top 10 words in positive reviews.

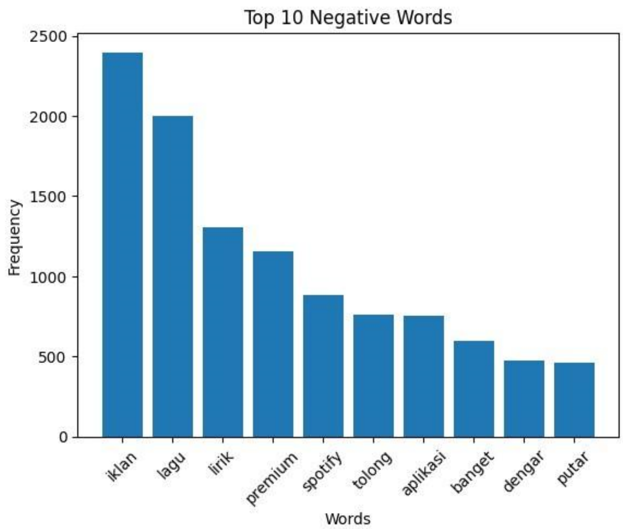


Figure 9. Top 10 words in negative reviews.

On the other hand, **Figure 9** displays the top 10 words that users usually use to express their negative sentiment in the review. These words are ‘iklan’, ‘lagu’, ‘lirik’, ‘premium’, ‘spotify’, ‘tolong’, ‘aplikasi’, ‘banget’, ‘dengar’, and ‘putar’. With almost 2500 frequencies, ‘iklan’ which translates to ‘ads’ appears on user’s negative reviews from the dataset. The word ‘iklan’ used by users in their negative reviews on the application. This indicates that the users’ main point of contemption are the ads for free users.

The positive and negative words review of Spotify users are also displayed in the word cloud generated by matplotlib in **Figures 10** and **11**. The word cloud visualizes words where the more frequently a word is used, the larger the font size it occupies. Words that are not displayed on the bar chart can be seen on the word cloud. **Figure**

11 shows the results of positive words that are used by users to express their positive sentiment while **Figure 10** shows the results of negative sentiment.

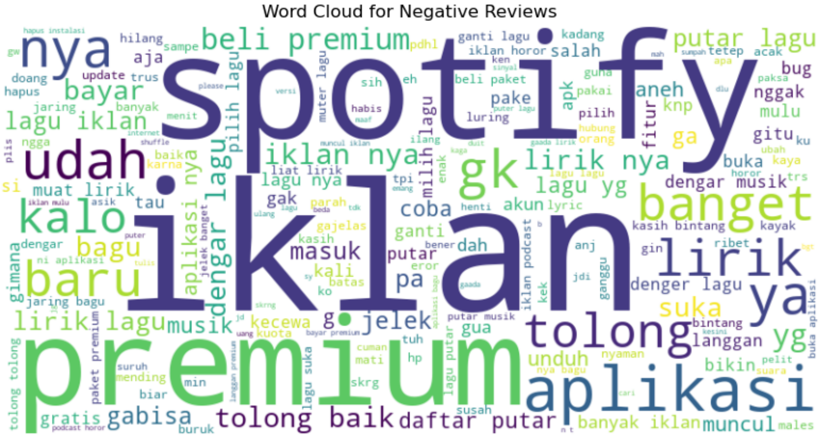


Figure 10. Word cloud in negative sentiment reviews.

Figure 11. Word cloud in positive sentiment reviews.

5.3. Sentiment analysis classification comparison

After evaluating the performance scores of these 3 classification models based on the precision, recall, *f*1-score, and accuracy data, the Support Vector Machine model exhibits the highest performance score, followed by Naive Bayes model and Random Forest. The difference of accuracy between the 3 models that were used in this research was not significant. The accuracy difference between SVM model with Naive Bayes and Random Forest model is 2 percent, while Naive Bayes and Random Forest have a very slight difference in accuracy, which is 0.001%. **Table 8** provides a more detailed breakdown of the performance scores of each model in percentage. The visual comparison of each model’s performance is provided in a clustered bar chart generated using matplotlib. Referring to **Figure 12**, the bar representing the Support Vector Machine model is slightly higher than the bar of the other model, indicating highest accuracy of that model.

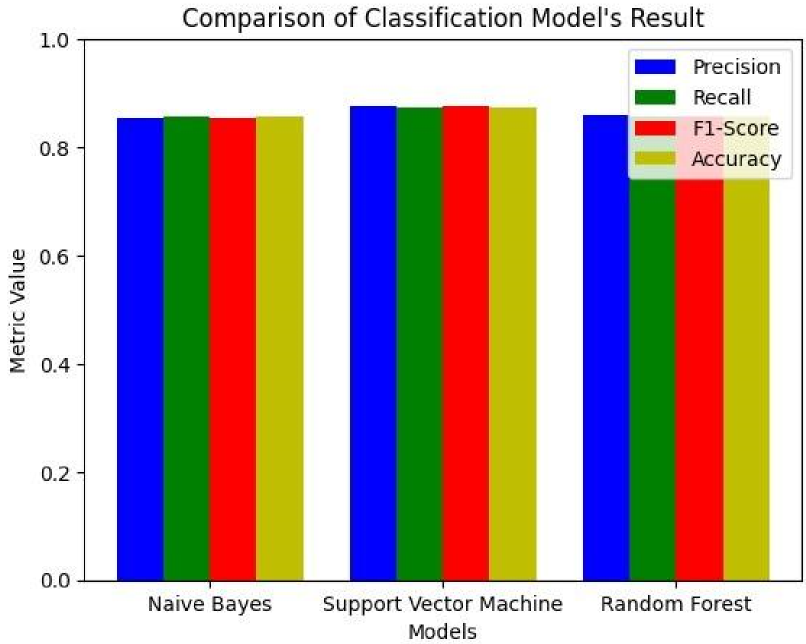


Figure 12. Classification model performance comparison.

Table 8. Comparison of classification model's result.

Model	Precision	Recall	*F*1-score	Accuracy
Naïve Bayes	0.854	0.857	0.854	0.857
Support vector machine	0.876	0.874	0.875	0.874
Random forest	0.859	0.856	0.857	0.856

There are 3 (three) models that were utilised in this research such as: Naïve Bayes, Support Vector Machine (SVM), and Random Forest. A variety of data mining and classification models were used to verify and evaluate the output of each model's effectiveness to this research which is Spotify review sentiment analysis.

5.3.1. Precision

Each model has its own precision value. The highest precision score result is owned by Support Vector Machine model with the value of 0.876, followed by Random Forest and Naïve Bayes, in comparison (Support Vector Machine > Random Forest > Naïve Bayes) with value of (0.876 > 0.859 > 0.854).

5.3.2. Recall

For recall score, the most suitable model is Support Vector Machine with accuracy value is 0.874, followed by Naïve Bayes and Random Forest model with slight difference between those. In comparison (Support Vector Machine > Naïve Bayes > Random Forest) with each value of (0.874 > 0.857 > 0.856).

5.3.3. F1-score

For *F*1-score, the most effective model is Support Vector Machine with accuracy value is 0.875, followed by Random Forest and Naïve Bayes model. In comparison (Support Vector Machine > Random Forest > Naïve Bayes) with each value of (0.875 > 0.857 > 0.854).

5.3.4. Accuracy

The most suitable model based on accuracy score is Support Vector Machine with an accuracy value of 0.874, followed by the Naïve Bayes and Random Forest model with slight difference between those. To compare (Support Vector Machine > Naïve Bayes > Random Forest) with value of (0.874 > 0.857 > 0.856).

In summary, after evaluating the sentiment analysis result on Spotify reviews, the Support Vector Machine is concluded to be the most suitable model for this paper as its results outperformed the other two in terms of accuracy across all comparisons.

5.4. Comparison of results

This study on sentiment analysis of Spotify user reviews utilized three prominent classification models: Naïve Bayes, Support Vector Machine (SVM), and Random Forest. The performance metrics of these models are summarized in **Table 8**.

The findings revealed that the Support Vector Machine achieved the highest performance with an $F1$-score of 0.875 and an accuracy of 0.874, outperforming both Naïve Bayes and Random Forest. This aligns with the prior findings in current literature where SVM and Random Forest frequently emerge as top-performing models. For instance, Madyatmadja et al. (2023) reported that SVM and Random Forest outperformed other algorithms like k-Nearest Neighbors and AdaBoost in classifying crowdsourced citizen complaints, with Random Forest achieving the highest accuracy. Similarly, Madyatmadja et al. (2022) demonstrated the superior performance of SVM and Random Forest in their analysis of user reviews of mutual fund applications, reaffirming the robustness of these models in various sentiment analysis contexts.

The results of Indulkar and Patil (2021) further support these findings. Though their study was conducted on Twitter, their comparative study of machine learning algorithms for sentiment analysis had also revealed that SVM and Random Forest provided the highest accuracy rates, highlighting similar results. Additionally, Velmurugan et al. (2022) identified SVM and Random Forest as the most effective classifiers in their sentiment analysis of customer reviews, emphasizing their consistency across different datasets and platforms. These studies collectively highlight the adaptability and high performance of SVM and Random Forest in sentiment analysis tasks.

In contrast, while Naïve Bayes demonstrated respectable performance in this study, it consistently ranked lower than SVM and Random Forest. This trend is evident in literature as well. For example, Varshney et al. (2020) found that ensemble techniques incorporating SVM and Random Forest outperformed Naïve Bayes in sentiment analysis of social networking sites. This study contributes to this growing body of evidence by not only confirming the superior performance of SVM and Random Forest but also providing a comprehensive comparison that underscores the importance of model selection in sentiment analysis. These insights are crucial for researchers and practitioners aiming to deploy effective sentiment analysis systems in real-world applications.

6. Conclusion

Spotify, the music world's leading streaming subscription-based application launched in 2008, has transformed how people enjoy and access music globally. It also has personalized recommendations and original content to each user that have made it a leader in the industry. As such, this research analyzed the sentiment of user reviews on Spotify, particularly focusing on Indonesians. With data extracted from the user reviews dataset from Google Play Store and Apps Store, this research conducted a sentiment analysis using the Naive Bayes, SVM, and Random Forest classification models.

Using 14,296 sample size of user reviews data, each model provided an impressive accuracy with the Support Vector Machine providing the highest accuracy of 87.4%. This was followed closely by the Naive Bayes and Random Forest model with accuracies of 85.7% and 85.6%, respectively. Compared to the obtained results, the synthesized literature review has revealed the potential direction for future research.

The quality of machine learning techniques such as Naive Bayes, SVM, and Random Forest is influenced by several critical factors. One significant factor is data preprocessing. The effectiveness of these models largely depends on the preprocessing steps applied to the data, including normalization, case folding, stop word removal, tokenization, and stemming. These techniques play a crucial role in ensuring the data is clean and standardized, which in turn enhances model performance. Another essential aspect is featuring selection. This study utilized Term Frequency-Inverse Document Frequency (TF-IDF) for feature extraction, emphasizing the importance of converting textual data into a numerical format that machine learning models can interpret. Additionally, the use of n-grams helps maintain context and understand word sequences, which are vital for analyzing complex patterns in the text.

Furthermore, model tuning is a critical factor that can significantly optimize the performance of each model. By adjusting hyperparameters, the models' accuracy and other evaluation metrics can be improved, leading to better overall performance. In this research, a comparison of the models demonstrated that the SVM model achieved the highest performance, with an $F1$-score of 0.875 and an accuracy of 0.874, surpassing both Naive Bayes and Random Forest. These results are consistent with prior findings in the literature, which frequently identify SVM and Random Forest as top-performing models in sentiment analysis tasks.

Directions for future work are based on the results of Support Vector Machine returning the highest accuracy rate for the dataset. For one, the use of this specific model in sentiment analysis could warrant further studies, as it is essential to test its validity and reliability on other data samples. As SVM has shown to have a reliable impact on sentiment analysis in various social media, it is worth investigating its reliability on further datasets, lest it is accurate but not generalisable.

Author contributions: Conceptualization, EDM, F, IGKE, JWI and DJMS; methodology, EDM and DJMS; software, F, IGKE and JWI; validation, F, IGKE and JWI; formal analysis, EDM, F, IGKE and JWI; investigation, F, IGKE and JWI; resources, EDM and DJMS; data curation, F, IGKE and JWI; writing—original draft

preparation, JWI; writing—review and editing, JWI; visualization, F and IGKE; supervision, EDM and DJMS; project administration, EDM and DJMS; funding acquisition, EDM. All authors have read and agreed to the published version of the manuscript.

Conflict of interest: The authors declare no conflict of interest.

References

Annur, C. M. (2023). Global Music Industry Revenue Dominated by Streaming Subscription by 2022 (Indonesian). Available online: https://databoks.katadata.co.id/datapublish/2023/03/27/pendapatan-industri-musik-global-didominasi-streaming-berlangganan-pada-2022 (accessed on 12 May 2024).

Aqlan, A. A., Manjula, B., & Lakshman Naik, R. (2019). A study of sentiment analysis: Concepts, techniques, and challenges. In: Proceedings of the International Conference on Computational Intelligence and Data Engineering. pp. 147–162. https://doi.org/10.1007/978-981-13-6459-4_16

Aufar, M., Andreswari, R., & Pramesti, D. (2020). Sentiment Analysis on YouTube Social Media Using Decision Tree and Random Forest Algorithm: A Case Study. In: Proceedings of the 2020 International Conference on Data Science and Its Applications (ICoDSA); Bandung, Indonesia. pp. 1–7. https://doi.org/10.1109/ICoDSA50139.2020.9213078

Dake, D. K., & Gyimah, E. (2023). Using sentiment analysis to evaluate qualitative students' responses. Educ Inf Technol, 28, 4629–4647. https://doi.org/10.1007/s10639-022-11349-1

Dr. Arivoli, & Sonali (2021). Sentiment Analysis Using Support Vector Machine Based on Feature Selection and Semantic Analysis. International Research Journal of Computer Science, 8(8), 209–214. https://doi.org/10.26562/irjcs.2021.v0808.009

Drus, Z., & Khalid, H. (2019). Sentiment Analysis in social media and its application: Systematic Literature Review. Procedia Computer Science, 161, 707–714. https://doi.org/10.1016/j.procs.2019.11.174

Indulkar, Y., & Patil, A. (2021). Comparative Study of Machine Learning Algorithms for Twitter Sentiment Analysis. In: Proceedings of the 2021 International Conference on Emerging Smart Computing and Informatics (ESCI); Pune, India. pp. 295–299. https://doi.org/10.1109/ESCI50559.2021.9396925

Irawanto, I., Widodo, C., Hasanah, A., et al. (2023). Sentiment Analysis and Classification of Forest Fires in Indonesia. ILKOM Jurnal Ilmiah, 15(1), 175–185. https://doi.org/10.33096/ilkom.v15i1.1337.175-185

Kumawat, S., Yadav, I., Pahal, N., et al. (2021). Sentiment Analysis Using Language Models: A Study. In: Proceedings of the 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence). pp. 984–988. https://doi.org/10.1109/Confluence51648.2021.9377043

Liu, Y., Han, T., Ma, S., et al. (2023). Summary of ChatGPT/GPT-4 Research and Perspective Towards the Future of Large Language Models. ArXiv, arXiv.2304.01852. https://doi.org/10.48550/arXiv.2304.01852

Madyatmadja, E. D., Shinta, Susanti, D., Anggreani, F., et al. (2022). Sentiment analysis on user reviews of mutual fund applications. Journal of Computer Science, 18(10), 885–895. https://doi.org/10.3844/jcssp.2022.885.895

Madyatmadja, E. D., Sianipar, C. P. M., Wijaya, C., et al. (2023). Classifying Crowdsourced Citizen Complaints through Data Mining: Accuracy Testing of k-Nearest Neighbors, Random Forest, Support Vector Machine, and AdaBoost. Informatics, 10(4), 84. https://doi.org/10.3390/informatics10040084

Mamun, M. M. R., Sharif, O. & Hoque, M. M. (2022). Classification of Textual Sentiment Using Ensemble Technique. SN Computer Science, 3(1). https://doi.org/10.1007/s42979-021-00922-z

Muhamad, N. (2023a). Spotify Premium subscribers reach 220 million by Q2 2023 (Indonesian). Available online: https://databoks.katadata.co.id/datapublish/2023/10/11/pelanggan-spotify-premium-tembus-220-juta-pengguna-pada-kuartal-ii-2023 (accessed on 12 May 2024).

Muhamad, N. (2023b). Spotify Premium Subscription Fee Increases, Here's How Much the New Plan Costs (Indonesian). Available online: https://databoks.katadata.co.id/datapublish/2023/07/27/biaya-langganan-spotify-premium-naik-ini-harga-paket-terbarunya (accessed on 12 May 2024).

Ogundare, O., & Araya, G. (2023). Comparative Analysis of CHATGPT and the evolution of language models. ArXiv, arXiv.2304.02468. https://doi.org/10.48550/arXiv.2304.02468

Singh, N. K., Tomar, D. S., & Sangaiah, A. K. (2020). Sentiment analysis: a review and comparative analysis over social media. Journal of Ambient Intelligence and Humanized Computing, 11(1), 97–117. https://doi.org/10.1007/s12652-018-0862-8

Spotify. (2023). Adjusting Our Spotify Premium Prices. Available online: https://newsroom.spotify.com/2023-07-24/adjusting-our-spotify-premium-prices/ (accessed on 12 May 2024).

Tripathy, A., & Rath, S. (2017). Classification of Sentiment of Reviews using Supervised Machine Learning Techniques. International Journal of Rough Sets and Data Analysis, 4(1), 56–74. https://doi.org/10.4018/ijrsda.2017010104

Usha, G., & Dharmanna, L. (2021). Sentiment Analysis on Business Data using Machine Learning. In: Proceedings of the 2021 Second International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE). https://doi.org/10.1109/icstcee54422.2021.9708593

Varshney, C. J., Sharma, A., & Yadav, D. P. (2020). Sentiment Analysis Using Ensemble Classification Technique. In: Proceedings of the 2020 IEEE Students Conference on Engineering & Systems (SCES); Prayagraj, India. pp. 1–6. https://doi.org/10.1109/SCES50439.2020.9236754

Velmurugan, T., Archana, M., & Latha, U. (2022). Sentiment Analysis of Customer Reviews Based Texts Using Classification Algorithms. In: Proceedings of the 8th Annual International Conference on Network and Information Systems for Computers (ICNISC); Hangzhou, China. pp. 185–191. https://doi.org/10.1109/ICNISC57059.2022.00047

Yadav, J. (2023). Sentiment Analysis on Social Media. Qeios. https://doi.org/10.32388/yf9x04

Yuan, J., Wu, Y., Lu, X., et al. (2020). Recent advances in deep learning–based sentiment analysis. Science China Technological Sciences, 63(10), 1947–1970. https://doi.org/10.1007/s11431-020-1634-3

Zhang, B. (2023). ChatGPT, an Opportunity to Understand More About Language Models. Medical Reference Services Quarterly, 42(2), 194–201. https://doi.org/10.1080/02763869.2023.2194149

Zheng, O., Abdel-Aty, M., Wang, D., et al. (2023). ChatGPT Is on the Horizon: Could a Large Language Model be Suitable for Intelligent Traffic Safety Research and Applications? ArXiv, arXiv.2303.05382. https://doi.org/10.48550/arXiv.2303.05382