Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 4, **Number** 2, November 2024
https://doi.org/10.47709/brilliance.v4i2.4877

Leveraging Machine Learning for Sentiment Analysis in Hotel Applications: A Comparative Study of Support Vector Machine and Random Forest Algorithms

Suryadi[1*], Dedek Syahputra [2], Nica Astrianda[3], Rizki Agam Syahputra[4], Rivansyah Suhendra[5]
[1,2,3,4,5]Universitas Teuku Umar, Aceh, Indonesia
[1]suryadi@utu.ac.id, [2]dedeksyahputra.tif20@gmail.com, [3]nicaastrianda@utu.ac.id, [4]rizkiagamsyahputra@utu.ac.id, [5]rivans.suhendra@gmail.com

open access

*Corresponding Author

Article History:
Submitted: 28-10-2024
Accepted: 29-10-2024
Published: 31-10-2024

Keywords:
Sentiment Analysis; User Reviews; Google Play Store; Support Vector Machine; Random Forest;

ABSTRACT

This research aims to conduct sentiment analysis on user reviews of hotel booking applications such as Trivago, Tiket, Booking, Traveloka, and Agoda, collected from the Google Play Store. The dataset used consists of 5,000 user reviews, with 80% of the data allocated for training and 20% for testing. Two algorithms applied in this study are Support Vector Machine (SVM) and Random Forest, with performance evaluation based on accuracy, precision, recall, and F1-score metrics. The test results show that the Random Forest algorithm delivers the best performance on the Trivago application with 94% accuracy, 94% precision, 100% recall, and a 97% F1-score. Random Forest proves to be more effective in handling diverse review data, while the Support Vector Machine (SVM) algorithm also produces good results in sentiment classification. This research contributes to the development of sentiment analysis based on user reviews, which can be utilized by app developers and hotel management to improve service quality and user experience.

INTRODUCTION

The hospitality industry is increasingly integrated with digital technology, where hotel booking applications like Trivago, Tiket, Booking, Traveloka, and Agoda have become essential tools for consumers in planning and reserving accommodations (Lam & Law, 2019). These applications allow users to compare prices, read reviews, and make reservations easily. User testimonials on platforms like Google Play provide valuable insights into customer experiences, covering aspects such as app usability, information accuracy, and the quality of hotel services booked through these platforms (Hardian et al., 2024). These reviews serve as a critical data source for improving service quality, benefiting both application developers and hotel management (Alhamdi, 2023).

User feedback on Google Play serves as an indicator of satisfaction or dissatisfaction with the services provided by both the app and the hotel. Using sentiment analysis, these testimonials can be analyzed to identify sentiment patterns—positive, negative, or neutral (Hardian et al., 2024). Sentiment Analysis techniques allow companies to systematically evaluate user feedback and extract insights useful for service improvement. In the context of hotel booking applications, such analysis is particularly relevant as it helps hotel management and app developers understand the service aspects most important to customers (Priyantina & Sarno, 2019).

To achieve accurate sentiment analysis results from user testimonials, selecting the appropriate algorithm is crucial. In this study, two algorithms are employed: Support Vector Machine (SVM) and Random Forest. SVM is a machine learning algorithm known for its effectiveness in text classification, especially when there is a clear margin between different sentiment categories. It operates by finding the optimal hyperplane that separates classes in the data with maximum margin (Khan et al., 2024). In contrast, Random Forest is an ensemble technique that combines multiple decision trees to enhance prediction accuracy and reduce overfitting risk, which is particularly valuable when handling large and diverse datasets like user reviews on Google Play (Khan et al., 2024).

This research utilizes a dataset containing user testimonials from Google Play for applications such as Trivago, Tiket, Booking, Traveloka, and Agoda. The primary goal of this study is to evaluate the performance of the SVM and Random Forest algorithms in performing Sentiment Analysis on user testimonials. Additionally, the research aims to identify trends and sentiment patterns that provide strategic insights for app developers and hotel management in improving service quality and user experience (Azzahra et al., 2020).

The findings of this study are expected to make a significant contribution to the field of sentiment analysis, particularly in the context of hotel booking applications. By gaining deeper insights into user sentiment, hotels and booking platforms can enhance their service quality, ultimately helping retain customers and increase competitive advantage in the global market.

E-ISSN : 2807-9035
Volume 4, **Number** 2, November 2024
https://doi.org/10.47709/brilliance.v4i2.4877

LITERATURE REVIEW

Research on sentiment analysis in the context of travel and hospitality applications has advanced significantly in recent years. A relevant study by Wardani and Ruldeviyani (2021) evaluated customer sentiment in reviews of Hotel XYZ on TripAdvisor to understand perceptions of the five-star hotel's service. Using a dataset of 160 reviews, they applied Decision Tree and Naive Bayes algorithms with k-fold cross-validation. The findings revealed that Decision Tree achieved higher accuracy and precision than Naive Bayes, though both models demonstrated high recall, indicating reliability in predicting positive sentiment. Word cloud visualizations showed that positive sentiments focused on room quality and staff service, while negative sentiments pertained to room conditions. The insights from this study provide hotel management with valuable information on improving service and identifying key areas of customer satisfaction. The study also suggests using reviews from multiple platforms for broader sentiment analysis in future research.

In another study, Nalawati et al. (2022) explored sentiment analysis on TripAdvisor hotel reviews using Named Entity Recognition (NER) to identify entities like location, facilities, and staff in user comments. A dataset of 600 reviews was collected through web scraping and classified as positive or negative using Naive Bayes Classifier. The preprocessing steps, including tokenization, stopword removal, and stemming, improved text accuracy. The combination of NER and Naive Bayes achieved peak accuracy of 91.67% with 90% training data, highlighting the correlation between dataset size and model accuracy. The data visualization in this study provides hotel management with insights into customer sentiment, aiding in service improvements based on customer feedback.

Additionally, the study by Pangestu et al. (2022) analyzes sentiment in hotel reviews on Agoda, focusing on Ibis Trans Studio Bandung, using the K-Nearest Neighbor (KNN) algorithm with Word2Vec feature extraction and a Skip-gram model. A total of 2,556 balanced reviews were collected and labeled as positive or negative sentiment. During processing, the study applied steps like stemming, stopword removal, and lemmatization to enhance model accuracy. Experiments showed that applying stemming and a higher vector dimension (300) in Word2Vec improved model performance, achieving the best accuracy of 82.61% and an F1 score of 83.96% with K=3 on KNN. The study concludes that adjusting vector dimensions and using stemming significantly impact model accuracy, recommending further exploration with neutral sentiment labeling and testing additional algorithms such as Naive Bayes, SVM**,** or Random Forest for comprehensive sentiment analysis.

Similarly, Dahlan et al. (2022) evaluated user sentiment regarding airline ticketing and hotel booking on the Traveloka app by analyzing Twitter reviews using Support Vector Machine (SVM). They collected data from Traveloka-related tweets posted between October and November 2022, resulting in 1,000 training samples labeled as positive or negative. The study applied preprocessing steps like cleaning, case folding, tokenizing, stopword removal, and stemming to improve text data quality. The analysis showed the SVM model achieved an accuracy of 75%, with a precision of 67%, recall of 100%, and an F1 score of 80%. This study highlights SVM's effectiveness in categorizing positive and negative sentiment in consumer reviews. It also recommends exploring alternative models and more specific feature extraction techniques to broaden sentiment coverage and support a more comprehensive analysis.

Further research by Chen et al. (2024) evaluated sentiment analysis of hotel reviews using a combined BERT and XGBoost model to enhance the accuracy of sentiment classification for positive and negative reviews. BERT was employed for feature extraction from the review text, leveraging the contextual word representations generated by the transformer architecture to effectively capture semantic information. Subsequently, these features were classified using XGBoost, a powerful gradient boosting algorithm suitable for classification tasks. The experiments used two public datasets, Ctrip_htl_2000 and Ctrip_htl_4000, containing 2,000 and 4,000 review samples respectively, with an equal distribution of positive and negative reviews. The experimental results showed that the combination of BERT and XGBoost achieved high accuracy, with F1-scores of 0.9200 and 0.9213 for both datasets. The study concluded that this approach outperformed other methods, such as Word2Vec with SVM or KNeighbors, in terms of accuracy and effectiveness for sentiment classification in hotel reviews, making it a reliable method for sentiment analysis in the hospitality industry.

Overall, these studies demonstrate that sentiment analysis is a highly valuable tool for understanding consumer opinions and has various practical applications in the travel and hospitality industries. The research also highlights the importance of selecting appropriate algorithms and adopting the latest technologies to improve the accuracy and effectiveness of sentiment analysis.

METHOD

In this study, the system workflow and stages were structured systematically through several main steps, as outlined in the work by Azzahra et al. (2020) in their sentiment analysis of hotel application reviews. The first step involves data collection through scraping, which enables the retrieval of raw data from digital platforms. This is followed by data labeling to classify each review as positive or negative, similar to the labeling process in Azzahra et al.'s research, helping to accurately identify sentiment categories.

Once labeled, data preprocessing is conducted to clean the dataset, involving steps such as tokenization, stopword removal, and stemming—techniques also utilized by Azzahra et al. Next, term weighting is applied to convert

Brilliance Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 4, Number 2, November 2024
https://doi.org/10.47709/brilliance.v4i2.4877

text into numerical representations, facilitating algorithmic processing. The processed data is then split into training and testing sets. Classification models are subsequently built using Support Vector Machine (SVM) and Random Forest algorithms, evaluated based on accuracy, precision, recall, and F1-score. This systematic approach closely follows the steps taken by Azzahra et al. (2020) in app-based sentiment analysis, reflecting methodological consistency across related studies. Each stage is comprehensively illustrated in Figure 1, outlining the workflow from start to finish.

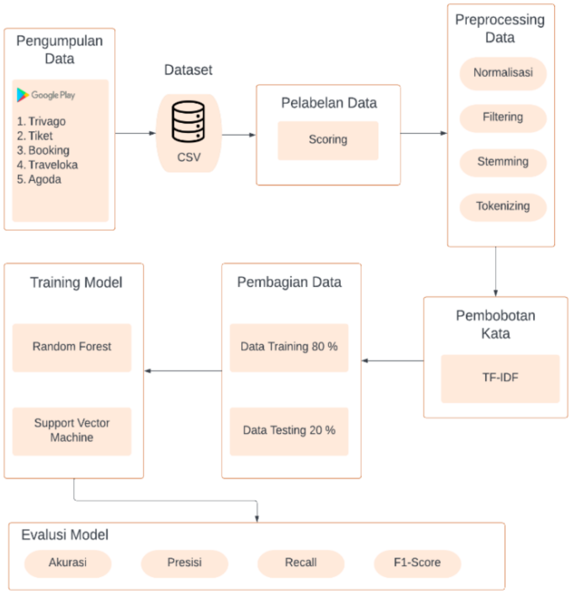

Figure 1. Research Methodology

Data Collection

The first step in the system workflow is data collection. This process is carried out using web scraping techniques, which enable automatic data retrieval from various online sources. In this study, scraping was performed by extracting comments or reviews from popular applications such as Trivago, Tiket, Booking, Traveloka, and Agoda on the Google Play platform, using the Google Play Scraper library. This library allows users to collect various data attributes, such as review title, review content, star rating, user name, and review publication date. The collected data, specifically the review content, is then used as the basis for building the sentiment analysis model.

Data Labeling

Data labeling in sentiment analysis is conducted automatically by utilizing the star ratings given by users in their reviews. Based on the study by Armykav et al. (2023), the sentiment labeling process using star ratings bears similarities to this research. In the study by Armykav et al. (2023), data were automatically labeled based on ratings, with reviews rated 4 and 5 stars categorized as positive sentiment, and ratings below 3 categorized as negative sentiment. Similarly, this study applies a comparable approach by designating ratings of 1 and 2 as the negative sentiment class, aiming to simplify the review labeling process based on overall user satisfaction.

Data Preprocessing

The data preprocessing process in this study shares similarities with the steps applied in the research by Amira et al., who employed similar techniques for sentiment analysis on hotel customer reviews. Amira et al. (2020) implemented preprocessing stages that include case folding, tokenizing, filtering, and stemming, aimed at refining raw data and preparing it to better suit the requirements of sentiment analysis using the SVM (Support Vector Machine) and Random Forest algorithms.

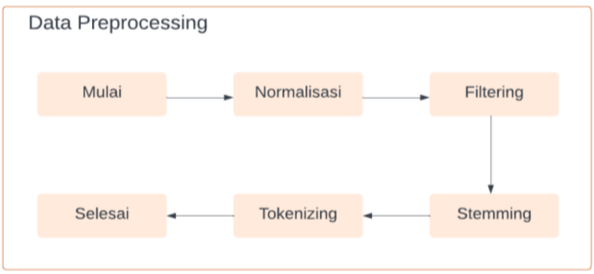

Figure 2. Flow Diagram for Preprocessing Steps

E-ISSN : 2807-9035
Volume 4, **Number** 2, November 2024
https://doi.org/10.47709/brilliance.v4i2.4877

Figure 2 illustrates the preprocessing steps applied to the dataset of reviews from Google Play, which includes applications such as Trivago, Tiket, Booking, Traveloka, and Agoda. The preprocessing steps undertaken in this study are as follows:

1. Cleaning
   The first step in text preprocessing is cleaning. This involves removing irrelevant or distracting elements from the text. The cleaning process includes eliminating non-alphanumeric characters such as punctuation marks, numbers, and symbols that do not have meaning in text analysis.
2. Case Folding
   The next step in preprocessing is case folding, which aims to convert all letters to lowercase. This step is important for maintaining data consistency, as in text analysis, words with uppercase and lowercase letters may be treated as different terms.
3. Tokenizing
   Tokenizing is the process of breaking down the text or reviews into smaller units of words or tokens. This step aims to decompose sentences or documents into words that can be analyzed separately. For example, the sentence "Saya suka membaca buku" would be tokenized into ["Saya", "suka", "membaca", "buku"]. Each token is then used in the feature extraction process using TF-IDF (Term Frequency-Inverse Document Frequency), which measures the importance of each word in the context of a specific document relative to the entire dataset.
4. Stopword Removal
   Stopword removal is a step in preprocessing that involves eliminating common words that do not have significant meaning in text analysis. Stopwords are words that frequently appear in reviews, such as "dan", "di", "ke", "yang", and "dengan", which typically do not contribute much to the meaning of the text in the context of sentiment analysis or text classification. Removing stopwords helps reduce noise and allows the focus to be placed on more relevant and meaningful words.
5. Stemming
   The final step in preprocessing is stemming. Stemming is the process of converting words to their root forms by removing affixes, such as prefixes and suffixes. For example, the word 'pelayanan " would be reduced to "layan". By simplifying word variations, stemming helps make the analysis more effective.

Word Weighting

The word weighting used in this study applies TF-IDF. TF-IDF (Term Frequency-Inverse Document Frequency) is a popular method in feature extraction for sentiment analysis that assesses the importance of words within a document. This process begins with calculating Term Frequency (TF), which is the frequency of a word's occurrence within a document, where frequently appearing words receive a higher TF value.

The next step is calculating Inverse Document Frequency (IDF), which measures how common or rare a word is across the entire document set. Words that appear in many documents will have a low IDF, while words that appear less frequently will have a high IDF, indicating the significance of that word. By multiplying the TF and IDF values, TF-IDF assigns weights to words based on their relevance in a particular document compared to the entire document set, so words with high TF-IDF weights are considered more relevant for sentiment analysis.

Data Splitting

In this study, the dataset is divided into two parts: 80% for training data and 20% for testing data. The training data is used to train the Support Vector Machine (SVM) algorithm and the Random Forest algorithm for classifying sentiments (positive and negative). The algorithms learn the sentiment values of each review to identify sentiment outcomes from reviews. After training the model, the remaining 20% of the dataset, which was not used in training, is used to test the model's performance. This evaluation includes measuring accuracy, precision, recall, and F1-score, so the model's ability to predict sentiment in new reviews can be assessed. This division ensures a balance between optimal training and proper evaluation.

Model Training

In the model training process, the algorithms used include Support Vector Machine (SVM) and Random Forest. SVM aims to find the optimal hyperplane to separate review sentiments, while Random Forest combines the results of multiple decision trees to achieve more accurate predictions. Both models learn sentiment patterns (positive, negative, or neutral) to accurately predict the sentiment of new reviews.

Model Evaluation

After training the model using training data, testing is performed to evaluate the performance of the Support Vector Machine (SVM) and Random Forest algorithms. This testing uses 20% of the total dataset set aside as testing data, which was not used during the training process. The testing data provides an objective assessment of the model's ability to predict sentiment in user reviews from applications like Trivago, Tiket, Booking, Traveloka, and Agoda. The

Brilliance Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 4, **Number** 2, November 2024
https://doi.org/10.47709/brilliance.v4i2.4877

evaluation methods include measurements of accuracy, precision, recall, and F1-score, used to assess how well the model classifies reviews into positive, negative, or neutral sentiment categories. Accuracy evaluates the model's overall performance, while precision and recall measure the model's accuracy and capability to identify relevant reviews. The results of this testing are used to determine which model performs best in sentiment analysis.

RESULT

In this stage, the research results are presented, covering the processes of data collection, data labeling, review preprocessing, word weighting, data distribution, sentiment analysis and model development, as well as model testing.

Data Collection

The dataset used in this research consists of user reviews from online hotel booking applications collected from the Google Play Store. These reviews were gathered using web scraping techniques, similar to the method used by Azzahra et al. (2020) in their sentiment analysis of hotel app reviews. A total of 1,000 reviews per application were used in the sentiment analysis process, resulting in a total of 5,000 reviews from the applications Trivago, Tiket, Booking, Traveloka, and Agoda. Figure 3 shows the dataset obtained from the Google Play Store.

	content	score
0	agoda semakin bagus harganya luar biasa, trmks...	5
1	entah ini apk legal atw bukan, ketika kami boo...	2
2	Rate yg ditampilkan lebih murah dr aplikasi la...	4
3	Ketika saya pesan antara harga yg tertera dan ...	2
4	Aplikasi scam, pesan hotel sudah bayar bukanya...	1
5	harga nya lebih murah di agoda yg bnyk diskonn...	4
6	Alhamdulillah uangnya sudah balik. Ternyata kl...	4
7	jangan mau deh pake Aplikasi ini, gw pesan tik...	1
8	Alhamdulilah ini perdana saya paksi agoda,,,ti...	5
9	saya sudah booking hotel dan sudah bayar, tapi...	2

Figure 3. Dataset from Google Play Store

Data Labeling

The data labeling process was conducted automatically based on the ratings provided by users. Ratings of 4 and 5 were categorized as positive class, while ratings of 1 and 2 were categorized as negative class. Figure 4 shows the labeled dataset according to its respective classes.

	content	score	Label
0	agoda semakin bagus harganya luar biasa, trmks...	5	Positif
1	entah ini apk legal atw bukan, ketika kami boo...	2	Negatif
2	Rate yg ditampilkan lebih murah dr aplikasi la...	4	Positif
3	Ketika saya pesan antara harga yg tertera dan ...	2	Negatif
4	Aplikasi scam, pesan hotel sudah bayar bukanya...	1	Negatif
5	harga nya lebih murah di agoda yg bnyk diskonn...	4	Positif
6	Alhamdulillah uangnya sudah balik. Ternyata kl...	4	Positif
7	jangan mau deh pake Aplikasi ini, gw pesan tik...	1	Negatif
8	Alhamdulilah ini perdana saya paksi agoda,,,ti...	5	Positif
9	saya sudah booking hotel dan sudah bayar, tapi...	2	Negatif

Figure 4. Dataset with Labelling Data.

Text Preprocessing

In this stage, text preprocessing is performed, which includes cleaning (data cleansing), case folding (converting all text to lowercase), tokenizing (breaking text into individual words), stopword removal (removing common, non-meaningful words), and stemming (converting words to their root forms).

1. Cleaning
   The cleaning process serves to remove special characters in comments, such as punctuation marks (e.g., commas, periods, question marks, exclamation marks), numeric characters (0-9), and other symbols (e.g., $, %, *). The results of removing these symbols are shown in Table 1.

E-ISSN : 2807-9035
Volume 4, **Number** 2, November 2024
https://doi.org/10.47709/brilliance.v4i2.4877

Table 1. Result of the Cleaning Process

Before	After
Comfortable spacious place. There is a mosque in front. Cool. Just suggest that there is a Qibla direction sign in the room.	A comfortable, spacious place with a cool mosque in front. Just make sure there is a sign indicating the direction of the Qibla in the room.

2. CaseFolding
Case folding is a step in text preprocessing that aims to convert all characters in the text to lowercase. This step is important because, in text analysis, differences between uppercase and lowercase letters are not considered significant. For example, the words "Traveloka," "TRAVELOKA," and "traveloka" are treated as the same entity after the case folding process. The result of case folding can be seen in Table 2.

Table 2. Result of Case Folding Process

Before	After
The negative comments about Traveloka are not true. I have been a Traveloka user for quite a long time and am satisfied with this application.	The negative comments about Traveloka are not true. I have been a Traveloka user for quite a long time and am satisfied with this application.

3. Tokenizing
Tokenizing is a step in text preprocessing that aims to break down text reviews into individual word segments. This process leverages the spaces within the comments to separate words. This step facilitates the process of calculating each word's weight using TF-IDF. The result of tokenizing can be seen in Table 3.

Table 3.Result of Tokenizing Process

Before	After
Ordering train tickets, plane tickets or hotels is easy and the prices are competitive	[‘book’, ’ticket’, ‘train’, ‘plane’, ‘hotel’, ‘easy’, ‘price’, ‘competitive’]

4. Stopword Removal
Stopword removal is a step in text preprocessing that aims to eliminate common or unimportant words in review classification, such as "and," "or," "therefore," "this," and "that," which hold no relevant meaning in text analysis. By removing stopwords, the text data becomes cleaner, allowing the analysis to focus on more meaningful words. This step helps reduce noise, thereby speeding up the analysis process and enhancing model performance. A stoplist dictionary, which is a collection of words considered unimportant, is used at this stage. If a document contains words matching those in the stoplist, the program removes them. The result of the stopword removal process can be seen in Table 4.

Table 4. Result of Stopword Removal Process

Before	After
[‘book’, ‘ticket’, ‘train’, ‘plane’, ‘or’, ‘hotel’, ‘so’, ‘easy’, ‘price’, ‘even’, ‘competitive’]	[‘book’, ’ticket’, ‘train’, ‘plane’, ‘hotel’, ‘easy’, ‘price’, ‘competitive’]

5. Stemming
Stemming is an important step in text preprocessing aimed at converting inflected words into their base form without affixes. This process removes prefixes, suffixes, or infixes, so words like "berjalan" (walking), "menjalani" (to go through), and "perjalanan" (journey) are converted to their root form "jalan" (road). The result of the stemming process can be seen in Table 5.

Table 5. Result of Stemming Process

Before	After
[‘help’, ‘travel’, ‘booking’, ‘ticket’, ‘easy’, ‘fast’, ‘application’]	[‘help’, ‘walk’, ‘order’, ‘ticket’, ‘easy’, ‘fast’, ‘application’]

Word Weighting

In this study, the Term Frequency - Inverse Document Frequency (TF-IDF) method is applied to determine the significance of words within a collection of hotel reviews from the Google Play Store. This approach utilizes two main parameters: the frequency of a word’s occurrence in a document (Term Frequency or TF) and the number of documents containing the word within the entire document collection (Inverse Document Frequency or IDF). Through TF-IDF, the relevance weight of each word can be identified for further analysis, such as text data clustering. Mishra et al. (2019)

E-ISSN : 2807-9035
Volume 4, **Number** 2, November 2024
https://doi.org/10.47709/brilliance.v4i2.4877

emphasize that TF-IDF is highly effective in extracting critical information from review text data, particularly in the context of sentiment analysis for hotel recommendations, where TF-IDF helps assign weights to words based on their occurrence in review documents.

1. Stage 1: Document Selection and Text Extraction
   The data used in this study comprises a collection of hotel reviews, divided into four documents, each containing distinct texts that reflect user experiences. Table 6 lists the documents used along with their corresponding reviews:

Table 6. Hotel Review Data

Document	Review
D1	clean comfortable room
D2	slow service not satisfactory
D3	affordable price spacious room
D4	bad slow wifi

2. Stage 2: Calculating Term Frequency (TF)
   In this stage, each word in the document is counted to produce the TF value, indicating how frequently the word appears in the respective document. For example, the word "room" appears once in document D1 and once in document D3 but does not appear in D2 or D4. The results of this calculation are shown in Table 7 below.

Table 7. Term Frequency Results

Term	TF (D1)	TF (D2)	TF (D3)	TF (D4)
Room	1	0	1	0
Clean	1	0	0	0
Comfortable	1	0	0	0
Service	0	1	0	0
Slow	0	1	0	1
Less	0	1	0	0
Satisfactory	0	1	0	0
Price	0	0	1	0
Affordable	0	0	1	0
Spacious	0	0	1	0
Wifi	0	0	0	1
Bad	0	0	0	1

3. Stage 3: Calculating Inverse Document Frequency (IDF)
   The next stage involves calculating the IDF, which indicates how rarely a word appears across the entire document collection. Words that appear frequently in multiple documents have a low IDF value, whereas words that appear infrequently have a higher IDF value, signifying a greater importance in the context of text interpretation. Table 8 below presents the IDF results for each unique word.

Table 8. Inverse Document Frequency Results

Term	DF	IDF
Room	2	0.30
Clean	1	0.60
Comfortable	1	0.60
Service	1	0.60
Slow	2	0.30
Less	1	0.60
Satisfactory	1	0.60
Price	1	0.60
Affordable	1	0.60
Spacious	1	0.60
Wifi	1	0.60
Bad	1	0.60

4. Stage 4: Calculating TF-IDF Value
   In the final stage, the TF and IDF values are multiplied to obtain the TF-IDF value for each word in a given document. For example, for the word "room" in D1, the TF-IDF value is calculated by multiplying the TF value of 1 by the IDF value of 0.30, resulting in a TF-IDF value of 0.30. The complete results of the TF-IDF calculations are shown in Table 9 below.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 4, **Number** 2, November 2024
https://doi.org/10.47709/brilliance.v4i2.4877

Table 9. Term Frequency - Inverse Document Frequency Results

Term	**IDF**	**TF (D1)**	**TF-IDF (D1)**	**TF (D2)**	**TF-IDF (D2)**	**TF (D3)**	**TF-IDF (D3)**	**TF (D4)**	**TF-IDF (D4)**
Room	0.30	1	0.30	0	0	1	0.30	0	0
Clean	0.60	1	0.60	0	0	0	0	0	0
Comfortable	0.60	1	0.60	0	0	0	0	0	0
Service	0.60	0	0	1	0.60	0	0	0	0
Slow	0.30	0	0	1	0.30	0	0	1	0.30
Less	0.60	0	0	1	0.60	0	0	0	0
Satisfactory	0.60	0	0	1	0.60	0	0	0	0
Price	0.60	0	0	0	0	1	0.60	0	0
Affordable	0.60	0	0	0	0	1	0.60	0	0
Spacious	0.60	0	0	0	0	1	0.60	0	0
Wifi	0.60	0	0	0	0	0	0	1	0.60
Bad	0.60	0	0	0	0	0	0	1	0.60

In the context of sentiment analysis, words with high TF-IDF weights can strongly indicate positive or negative sentiment in hotel reviews. Words such as "clean" and "comfortable" have high weights in positive reviews, while words like "slow" and "bad" stand out in negative reviews. These results allow the model to recognize word patterns more accurately, thereby enhancing the precision of sentiment predictions for hotel reviews on the Google Play Store.

Data Splitting

In this study, the number of user review data points used for each application—Trivago, Tiket, Booking, Traveloka, and Agoda—is 1,000 per application. The data distribution consists of 800 data points for training, with 400 in the positive class and 400 in the negative class. Meanwhile, 200 data points are allocated for testing. The data split results are shown in Table 10.

Table 10. Data Distribution

No	**Application Name**	**Training Data**		**Testing Data**	**Total Data**
		Positive Data	Negative Data		
1.	Trivago	400	400	200	1000
2.	Tiket	400	400	400	1000
3.	Booking	400	400	400	1000
4.	Traveloka	400	400	400	1000
5.	Agoda	400	400	400	1000

Training Data Modeling

The modeling in this study utilizes the Support Vector Machine and Random Forest algorithms.

1. Support Vector Machine
   Support Vector Machine (SVM) is a machine learning algorithm that operates by identifying the optimal hyperplane to separate data classes, in this case, reviews with positive and negative sentiment. This algorithm aims to find the largest separating margin between the two classes, thereby enhancing classification performance on data with distinct margins. In the context of sentiment analysis for hotel booking application reviews, SVM is particularly effective when there is a significant difference between positive and negative sentiments. For instance, reviews consistently using terms such as 'sangat bersih,' 'nyaman,' or 'ramah' are generally easy to separate from reviews with terms like 'kotor', 'bising', or 'tidak memuaskan'. However, challenges arise when reviews exhibit mixed sentiments, such as 'great location but slow service.' In such cases, data cannot always be separated by a linear boundary, leading to decreased SVM performance due to difficulties in identifying the optimal margin. Consequently, in situations where the boundary between two classes is unclear, SVM tends to experience a decline in accuracy compared to other algorithms that are more flexible in handling complex data. Simarmata et al. (2023) stated that SVM consistently demonstrates high performance with an accuracy rate of 94% in sentiment analysis of hotel reviews, highlighting its effectiveness on data with clearly separated sentiments.
2. Random Forest
   The Random Forest algorithm employs an ensemble learning approach, which combines decisions from multiple classification trees to achieve a more accurate and stable final outcome. Each decision tree within the Random Forest model is built from random subsets of data and features, collectively introducing diversity in sentiment classification. This process plays a crucial role in enhancing the model's accuracy and stability, particularly when dealing with data characterized by diverse and often inconsistent patterns. In the context of sentiment analysis, for instance, a review stating, 'kamar nyaman tapi wifi lambat' may lead some decision trees to interpret room comfort

E-ISSN : 2807-9035
Volume 4, **Number** 2, November 2024
https://doi.org/10.47709/brilliance.v4i2.4877

as a positive sentiment, while others emphasize the Wi-Fi complaint as a negative sentiment. By aggregating the results of individual trees through a voting process, Random Forest produces a more robust decision, excelling in handling data containing both positive and negative elements. According to Baskoro et al. (2021), this technique significantly improves the accuracy of sentiment analysis models compared to other methods, such as SVM or Naïve Bayes, due to the ensemble's ability to integrate diverse perspectives from each tree.

Model Testing

Model testing in this study uses the Support Vector Machine and Random Forest algorithms, with performance measured based on accuracy, precision, recall, and F1-score. Table 11 shows the confusion matrix results using the Support Vector Machine (SVM) algorithm, based on a data split of 80% for training and 20% for testing. Out of 200 reviews used as testing data, the results indicate that the best performance was obtained with the Trivago app, achieving an accuracy of 94%, precision of 93%, recall of 100%, and F1-score of 97%. Consistent with the study by Azzahra et al., which found that the Naive Bayes method achieved a highest accuracy of 94% on the Trivago app, these results indicate good performance in sentiment classification for hotel booking app reviews from the Google Play Store (Azzahra et al., 2020).

Table 11. Confusion Matrix Using Support Vector Machine

No	Application Name	SVM (Support Vector Machine)			
		Akurasi	Presisi	Recall	F1-Score
1.	Trivago	94 %	93 %	100 %	97 %
2.	Tiket	80 %	89 %	71 %	79 %
3.	Booking	80 %	91 %	67 %	77 %
4.	Traveloka	74 %	78 %	63 %	70 %
5.	Agoda	77 %	78 %	75 %	77 %

Model testing with the Random Forest algorithm also involves measuring accuracy, precision, recall, and F1-score. Table 12 presents the confusion matrix results using the Random Forest algorithm, based on an 80% split for training and 20% for testing. Out of a total of 200 reviews used as testing data, the results show that the best performance was obtained with the Trivago app, achieving an accuracy of 94%, precision of 94%, recall of 100%, and F1-score of 97%.

Table 12. Confusion Matrix Using Random Forest

No	Application Name	Random Forest			
		Akurasi	Presisi	Recall	F1-Score
1.	Trivago	94 %	94 %	100 %	97 %
2.	Tiket	79 %	88 %	70 %	78 %
3.	Booking	80 %	89 %	70 %	78 %
4.	Traveloka	76 %	76 %	71 %	73 %
5.	Agoda	76 %	75 %	78 %	76 %

CONCLUSION

The conclusion contains a summary of what is learned from the results obtained, what needs to be improved in further study. Other common features of the conclusions are the benefits and applications of the research, limitation, and recommendations based on the results obtained. Based on the research results, it can be concluded that the Support Vector Machine (SVM) and Random Forest algorithms were successfully employed in sentiment analysis of user reviews for hotel booking applications such as Trivago, Tiket, Booking, Traveloka, and Agoda. Using a dataset of 5,000 reviews with an 80% split for training and 20% for testing, the best performance was achieved with the Trivago app, reaching an accuracy of 94%, precision of 94%, recall of 100%, and F1-score of 97%. The Random Forest algorithm proved capable of handling substantial and diverse data variations effectively, while SVM was efficient in classifying sentiment with clear margins.

Overall, both algorithms delivered satisfactory results; however, Random Forest demonstrated superior capability in managing complex review datasets. This research offers valuable insights for app developers and hotel management to enhance services based on user sentiment analysis.

REFERENCES

Amira, S. A., Utama, S., & Fahmi, M. H. (2020). *Penerapan Metode Support Vector Machine untuk Analisis Sentimen pada Review Pelanggan Hotel*. Edu Komputika, 7(2), 40-48. Universitas Negeri Semarang.

Armykav, R., Mantoro, T., Ayu, M. A., & Asian, J. (2023). *Sentiment Analysis CNN Indonesia App Reviews on Play Store Using Naive Bayes Algorithm*. *International Conference on Technology, Engineering, and Computing Applications (ICTECA)*, IEEE, pp. 1-5.

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 4, **Number** 2, November 2024
https://doi.org/10.47709/brilliance.v4i2.4877

Baskoro, B. B., Susanto, I., & Khomsah, S. (2021). *Analisis Sentimen Pelanggan Hotel di Purwokerto Menggunakan Metode Random Forest dan TF-IDF (Studi Kasus: Ulasan Pelanggan Pada Situs TripAdvisor)*. Journal of Informatics, Information System, Software Engineering and Applications, 3(2), 21-29.

Lam, C., & Law, R. (2019). Readiness of upscale and luxury-branded hotels for digital transformation. *International Journal of Hospitality Management*, 79, 60-69.

Alhamdi, R. (2023). Pengaruh online review dan harga terhadap keputusan pemesanan kamar hotel di online travel agent (Studi kasus kota Batam). *Jurnal Manajemen Perhotelan*, 9(2), 63-70.

Azzahra, Z. F., Andreswari, R., & Hasibuan, M. A. (2020). Sentiment analysis website of online hotel booking application reviews using the Naive Bayes algorithm. In *2020 6th International Conference on Science and Technology (ICST)*, Yogyakarta, Indonesia (pp. 1-6).

Chen, M., Xu, H., Wu, Y., & Wu, J. (2024). *Sentiment Analysis of Hotel Reviews based on BERT and XGBoost. International Conference on Computer Technologies (ICCTech).*

Hardian, R., Oktaviana, L. D., & Hamdi, A. (2024). Sentiment analysis of pegipegi.com on Google Playstore with Naïve Bayes algorithm. *JURTEKSI (Jurnal Teknologi dan Sistem Informasi)*, 10(3), 583–590.

Khan, T. A., Sadiq, R., Shahid, Z., Alam, M. M., & Su'ud, M. B. M. (2024). Sentiment analysis using Support Vector Machine and Random Forest. *Journal of Informatics and Web Engineering*, 3(1), 67-75.

Mishra, R. K., Urolagin, S., & Jothi, A. A. (2019). *A Sentiment analysis-based hotel recommendation using TF-IDF Approach*. In *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)* (pp. 811-815).

Nalawati, R. E., Liliana, D. Y., Nugrahani, F., Abiyanka, F. H., & Karrel, R. (2022). *Sentiment Analysis on Tripadvisor Hotel Review using Named Entity Recognition. International Conference on Information and Communications Technology (ICOIACT).*

Pangestu, V. C., Adiwijaya, & Purbolaksono, M. D. (2022). *Sentiment Analysis on Hotel Review in Bandung from Website Agoda Using KNN Algorithm. International Conference on Software Engineering and Information Technology (ICoSEIT).*

Priyantina, R. A., & Sarno, R. (2019). Sentiment analysis of hotel reviews using latent Dirichlet allocation, semantic similarity, and LSTM. *International Journal of Intelligent Engineering and Systems*, 12(4), 130-136.

Simarmata, A. R., & Zakariyah, M. (2023). *Sentiment Analysis of Hotel Reviews Using Support Vector Machine*. Indonesian Journal of Computer Science, 12(5), 2603-2609.

Wardani, S. K., & Ruldeviyani, Y. (2021). *Sentiment Analysis of Visitor Reviews on Hotel in West Sumatera. International Workshop on Big Data and Information Security (IWBIS).*