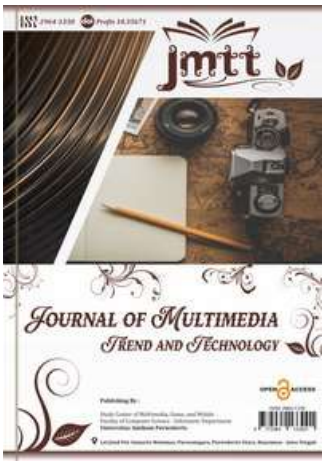
Journal of Multimedia Trend and Technology - JMTT
Volume 4, Issue 2, August 2025, ISSN 2964-1330
https://journal.educollabs.org/index.php/jmtt/

Comparison of Support Vector Machine and XGBoost Algorithms in Sentiment Analysis of Visitor Reviews of Baturraden Tourism Forest

Catur Risma Utami[1*], Irfan Santiko[2]

[1]Information System Department, Faculty of Computer Science, Universitas Amikom Purwokerto, Indonesia
[2]Informatic Department, Faculty of Computer Science, Universitas Amikom Purwokerto, Indonesia

ARTICLE INFO



History :

Submit on 20 May 2025
Review on 10 June 2025
Accepted on 23 July 2025

Keyword :

Sentiment Analysis;
Support Vector Machine;
XGBoost;
Google Maps;
Tourist Attractions

ABSTRACT

The Google Maps platform provides a platform for visitors to express their opinions through reviews. This study aims to compare the performance of the Support Vector Machine and XGBoost algorithms in sentiment analysis of Baturraden Tourism Park visitor reviews. Data were collected using scraping techniques and obtained 4,096 reviews. After going through preprocessing stages including cleaning, tokenization, normalization, stopword removal, and stemming, the data used in the analysis process amounted to 2,912 reviews. Word weighting was carried out using the TF-IDF method, and the SMOTE technique was applied to address class imbalance. The results showed that the Support Vector Machine algorithm performed better than XGBoost with an accuracy rate of 94.52% before SMOTE and 94.86% after SMOTE, while XGBoost obtained an accuracy of 92.80% before SMOTE and 93.15% after SMOTE. These findings indicate that the Support Vector Machine is more effective in classifying positive and negative sentiments. This study is expected to contribute to the application of machine learning methods to understand visitor opinions on the Google Maps platform. Especially in the context of tourist attractions.

***Corresponding Author:**

Catur Risma Utami
Department of Informatics, Faculty of Computer Science, Universitas Amikom Purwokerto, Banyumas, Central Java, Indonesia.
Email: *risma.utami@gmail.com*

INTRODUCTION

Tourism is a crucial sector in the Indonesian economy. The tourism industry has grown rapidly in recent years, demonstrating its global popularity. According to Statista, revenues in the travel and tourism sector are expected to reach US$854.40 billion by 2023, with an estimated annual revenue growth rate of 4.41% (2023-2027), generating a market volume of US$1,016.00 billion by 2027 [1][2].

Banyumas Regency, located in Central Java Province, is known for its natural and cultural riches. One of its main attractions is Baturraden. Baturraden, a sub-district in Banyumas Regency, boasts a number of tourist attractions [3]. One such attraction is the Baturraden Forest Tourism Park, which boasts attractions such as the Baturraden Botanical Gardens, Pancuran Pitu, and the Labyrinth Garden.

With the advancement of information technology, tourists now have easier access to share their experiences through reviews and comments on various platforms, one of which is Google Maps [4][5]. Google Maps is an internet-based guide application provided free of charge by Google. Users leave reviews in the form of text and ratings, written reviews containing real experiences, opinions, as well as the quality of the destination, facilities and overall experience [6][7]. This shows that online public opinion plays a crucial role in shaping tourists' perceptions of tourist attractions, so destination managers must understand and utilize this data to improve tourism management and services [8]. The large number of reviews available online presents new challenges for tourism destination managers. Thousands of review data are difficult to analyze manually because they are unstructured and written in random text [9][10].

Google Maps is a free online mapping application provided by Google. It can be accessed through a web browser or mobile device, making it convenient for users in a variety of situations. In addition to serving as a navigation platform, this platform also allows users to search for information about a location. Google Maps also provides additional features such as reviews of tourist attractions [18][19].

Reviews that describe various experiences and opinions from visitors regarding service, cleanliness, facilities, and visitor satisfaction often do not align with the listed ratings. Reviews contain sentiment that can be used as a tool to measure satisfaction levels and identify areas for improvement [20][21]. The large volume of review data makes manual analysis very difficult. Without a systematic classification process for visitor sentiment, the evaluation process becomes slow, inefficient, and can potentially lead to subjective bias due to reliance on personal judgment [22].

If this information is not processed properly, potentially useful suggestions and criticisms from visitors will be overlooked [8][9]. Therefore, a computational analysis approach such as sentiment analysis is needed to automatically identify positive, negative, and neutral opinions from reviews, so that the results can be used as a basis for evaluating the management and improving the quality of tourist destinations. Sentiment analysis is a part of Natural Language Processing (NLP) which functions as an effective method for collecting and evaluating opinions from several people or communities, usually through comments or writing [11][12][13]. This technique allows for the extraction of information from unstructured data, where opinions can be categorized as positive or negative sentiment based on the comments given [14].

Previous research on sentiment analysis has been widely used to process visitor reviews on digital platforms [15]. However, selecting the most appropriate classification algorithm remains challenging due to the differing characteristics of text data. Support Vector Machine and XGBoost algorithms are frequently used in sentiment classification, but they have different learning approaches, potentially yielding different performance results on the same dataset [16][17]. Therefore, this study is needed to compare the performance of Support Vector Machine and XGBoost in classifying visitor review sentiment.

Based on the description above, this study was conducted to compare the performance of Support Vector Machine and XGBoost algorithms in classifying visitor review sentiment at Wana Wisata Baturraden on Google Maps using TF-IDF weighting. Reviews were categorized into two classes: positive and negative sentiment. The results of this study indirectly provide an objective overview of the performance of each algorithm and serve as evaluation material for managers in understanding visitor opinions. This research not only contributes to tourism management but also provides a reference for selecting an appropriate machine learning algorithm for sentiment analysis in tourism reviews.

METHOD

This research began with data collection on visitor review sentiment at *Wana Wisata Baturraden* through the apify.com platform using the scraping method. After data collection, the next step was manual labeling, followed by data processing, which involved several important procedures, including cleaning, tokenizing, normalization, stop-word removal, and stemming. This process aimed to improve data quality before conducting more in-depth analysis.After completing the preprocessing stage, the dataset was separated into two segments: training data and testing data, to ensure effective model testing. After the data splitting process was complete, the next stage was TF-IDF word weighting. To address imbalances in the training data, oversampling was performed. Furthermore, in the modeling stage, the Support Vector Machine and XG-Boost algorithms were used with the help of Google Collaboratory to assess the performance of each algorithm in categorizing user sentiment. The final stage was evaluating the results using a Confusion Matrix to measure the performance of both algorithms. An illustration of this research concept can be seen in Figure 1 below:

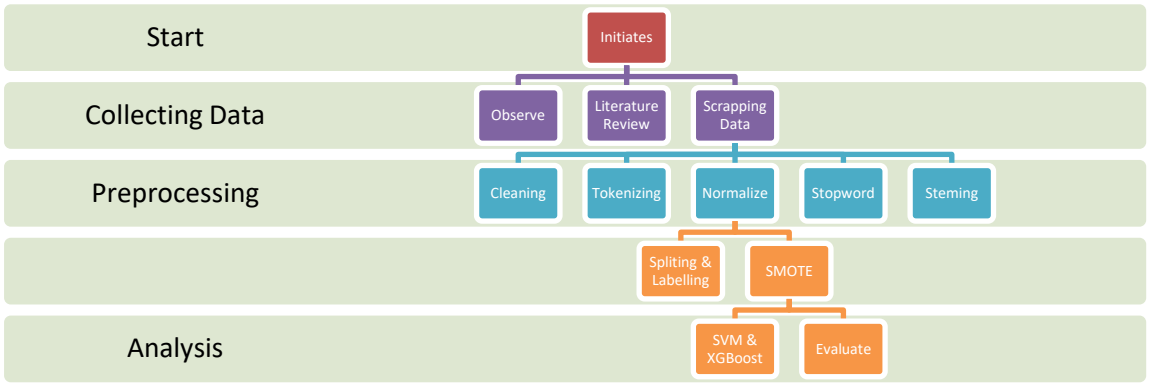


Figure 1. Research Concept

1. Cleaning is the initial step taken to remove irrelevant data from repetitive text columns and unnecessary attributes. This stage aims to remove numbers, certain symbols, excessive spaces, punctuation, emojis, and repeated characters within the sentence content.
2. Tokenization is the process of dividing text or sentences into smaller parts called tokens. Tokens can include words, phrases, or other entities that have meaning within the context of the text. The goal is to facilitate data analysis so that it can be analyzed more efficiently by natural language processing algorithms or other tools.
3. Normalization is a crucial step in text preprocessing, aiming to transform non-standard words in a data set into correct words that comply with applicable language rules. The purpose of the text normalization process is to eliminate uniform representations of various variations in the data or text, thus ensuring consistency in data evaluation.
4. Stopword Removal or Filtering is the step carried out to remove unnecessary words or phrases to minimize the number of words stored by the system.
5. Stemming is the step of converting words with affixes into root words. The goal is to minimize word variation in documents, making patterns easier to recognize and increasing the accuracy of sentiment analysis.
6. Data labeling is used to categorize data according to its sentiment type, which aims to identify whether the data falls into the positive, negative, or neutral sentiment group. However, in this study, neutral sentiment will be omitted, resulting in only positive and negative sentiments. The data labeling process in this study uses a lexicon-based technique.
7. This study uses two types of data: training and testing data. Training data is a collection of labeled data that will be processed using the method used to produce a pattern or model as a prediction for the testing data. Meanwhile, testing data is data used to test the performance of the classification model. Testing data has categories used to test the model's accuracy in classifying

data. Proportions of 60:40, 70:30, 80:20, and 9:10 were used to divide the training and testing data [22][17][19].

8. At this stage, each word in the document is assigned a value using the Term Frequency-Inverse Document Frequency (TF-IDF) technique. The TF-IDF method calculates term frequency (TF) and inverse document frequency (IDF) values for each word in a document within a word frame. Term Frequency (TF) measures how frequently a word appears in a document, and Document Frequency (DF) measures the number of documents containing a particular word. Inverse Document Frequency (IDF) is the weighted inverse of the Document Frequency weight.
9. SMOTE, or data balancing, is an oversampling method that attempts to balance the dataset by generating synthetic minority instances. Imbalance occurs when one sentiment class, such as positive or negative, has a different amount of data, either more or less than the other class. If this imbalance is not addressed, it can impact the model's effectiveness in recognizing patterns proportionally across all categories. Therefore, this technique is used to increase the amount of data in the underserved class, thus ensuring a more even data distribution and allowing the model to train more effectively.

The Support Vector Machine application is used to build a sentiment classification model by finding the optimal hyperplane capable of maximally separating data between classes. The classification process is carried out by maximizing the distance (margin) between data from different classes, so the model is able to produce a more optimal and accurate class separator, especially for high-dimensional data such as TF-IDF feature extraction results. The eXtreme Gradient Boosting (XG-Boost) algorithm is applied as a comparison method in sentiment classification. XG-Boost works by building a model incrementally to correct previous prediction errors. This algorithm is known for its high computational speed and good generalization capabilities, making it effective in improving classification accuracy. The stage after modeling is evaluation to assess the performance of the created model. The purpose of the evaluation is to measure the model's accuracy in classification. To assess model performance, we used the Confusion Matrix, which measures various evaluation metrics such as accuracy, precision, recall, and f1-score. These metrics provide in-depth insights into how well the model accurately classifies sentiment and how efficiently it handles varied data.

RESULT

A. Data Collecting.

This stage involved collecting review data using web scraping for the Wana Wisata Baturraden tourist attraction on Google Maps. The web scraping process used the apify.com platform, which generated data in CSV format. A total of 4,096 reviews were collected, with the most relevant review categories. Data was collected in October 2024. An example of a visitor review of the Wana Wisata Baturraden tourist attraction using keywords is shown in Figure 2 below.

Figure 2. Data Collection Results

B. Data Preprocessing.

The preprocessing stage consists of several processes, because reviews usually contain a lot of noise and tend to have an irregular structure. The goal of preprocessing is to clean and tidy up raw data so that it can be processed further easily. By performing preprocessing, researchers can remove distracting elements in the text. Before preprocessing, there are libraries that need to be prepared. In this study, Python libraries were used to support the text preprocessing process. The NumPy library was used to assist with numeric data processing, while Pandas was used to read, manage, and manipulate review data in tabular form. In addition, the re library was used to clean text using regular expressions, such as removing unnecessary characters, while the string library was used to assist in the process of cleaning punctuation in the text.

During the data cleaning phase, a library called re (regular experiment) needs to be prepared. The re library plays a crucial role in the data cleaning process, specifically in removing irrelevant characters before conducting in-depth analysis in sentiment classification. The re library plays a significant role in transforming raw data into clean and more accurate text, thereby improving the accuracy and efficiency of machine learning models in identifying sentiment patterns. The code for implementing the cleaning process is as follows:

```
import re

def clean_text(text):
    if isinstance(text, str):
        text = text.lower()  # Ubah semua jadi huruf kecil (case folding)
        text = re.sub(r'http\S+', '', text)  # Hapus URL
        text = re.sub(r'\d+', '', text)  # Hapus angka
        text = re.sub(r'[^a-z\s]', '', text)  # Hapus tanda baca & simbol
        text = re.sub(r'(.)\1{2,}', r'\1', text)  # Hapus huruf berulang
        text = re.sub(r'\s+', ' ', text)  # Hapus spasi ganda di tengah
        text = text.strip()  # Hapus spasi di awal & akhir
        return text
    else:
        return text
```

Figure 3. Cleaning Stage Script

Using the clean_text function, all text in the text column will be cleaned of irrelevant elements. The steps in this process include:

1. Converting all letters in the text to lowercase to standardize word forms.
2. Removing URLs or links that begin with "https://."
3. Removing all numbers in the text.
4. Removing unclear characters other than letters and spaces (such as symbols or punctuation).
5. Removing repeated letters, such as the word "baguuuus" becoming "bagus."
6. Removing extra spaces to ensure clean and neat text formatting.

The results of the cleaning process using Google Collaboratory can be seen in the text in Table 4.2 below.

Table 1. Example of Cleaning Process Results

Before	After
Masuk lewat kebun raya Baturaden, lewatin bobocabin dll, jalan terus tp jalannya lumayan bagus siy, pas deket2 lokasi pintu gerbang jalur pendakian baru mulai gerasak geruduk jalan nya. Sampe lokasi parkir wow banget, hutan, bayar seikhlasnya, kita pakai motor kasih 5rb, tp pas pulang kang parkir dah gada wkwkw soale dah mulai gelap juga, kabut udh turun n sepi, buka sampe jam 4 aja. Dari parkiran masih jalan kaki 300 meteran turun tp jalannya bagus udh bertangga, kalau capek bisa berenti di gazebo 2 di pinggiran ada di bbrp titik. Ga bawa baju ganti? Aman, banyak yg jual di lokasi, ada makanan n minuman juga dan souvenir. Kita pesen Onsen private harga 150rb buat 1jam, cukup buat 5 orang, ada kamar mandi bilasnya, tp sayang Krn dah agak sore air kamar mandi udh mati nyala hehe... Tp air panas n view nya worthed banget, sewa handuk nya 10k. Pr nya siy pas balik ke atas mendaki banget wkwkw... Tp semuanya seru dan worthed banget	*masuk lewat kebun raya baturaden lewatin bobocabin dll jalan terus tp jalannya lumayan bagus siy pas deket lokasi pintu gerbang jalur pendakian baru mulai gerasak geruduk jalan nya sampe lokasi parkir wow banget hutan bayar seikhlasnya kita pakai motor kasih rb tp pas pulang kang parkir dah gada wkwkw soale dah mulai gelap juga kabut udh turun n sepi buka sampe jam aja dari parkiran masih jalan kaki meteran turun tp jalannya bagus udh bertangga kalau capek bisa berenti di gazebo di pinggiran ada di bbrp titik ga bawa baju ganti aman banyak yg jual di lokasi ada makanan n minuman juga dan souvenir kita pesen onsen private harga rb buat jam cukup buat orang ada kamar mandi bilasnya tp sayang krn dah agak sore air kamar mandi udh mati nyala hehe tp air panas n view nya worthed banget sewa handuk nya k pr nya siy pas balik ke atas mendaki banget wkwkw tp semuanya seru dan worthed banget*

Before	After
Tempatnya asik, dingin dan kadang berkabut. Pemandangan bagus khas pegunungan. *Perlu fisik yg bagus untuk mendaki tangga pulangnya.* *Akses jalan menuju lokasi ada yg rusak.* *Toilet di pintu masuk kurang memadai.*	*tempatnya asik dingin dan kadang berkabut pemandangan bagus khas pegunungan perlu fisik yg bagus untuk mendaki tangga pulangnya akses jalan menuju lokasi ada yg rusak toilet di pintu masuk kurang memadai*

The tokenization process is one of the important stages in text preprocessing that aims to break down text into smaller, more meaningful units, called tokens. The required library, namely the NLTK library, is used in the tokenization stage. The word_tokenize() function utilizes the punct resource to recognize word boundaries and punctuation so that the text can be further processed in the sentiment analysis stage. script for the tokenizing stage. The tokenizing() function is used to ensure that the tokenization process is only applied to string-type data, thus avoiding processing errors on invalid data. The Tokenizing example is shown in Table 2 below.

Table 2. Example of Tokenization Results

Before	After
masuk lewat kebun raya baturaden lewatin bobocabin dll jalan terus tp jalannya lumayan bagus siy pas deket lokasi pintu gerbang jalur pendakian baru mulai gerasak geruduk jalan nya sampe lokasi parkir wow banget hutan bayar seikhlasnya kita pakai motor kasih rb tp pas pulang kang parkir dah gada wkwkw soale dah mulai gelap juga kabut udh turun n sepi buka sampe jam aja dari parkiran masih jalan kaki meteran turun tp jalannya bagus udh bertangga kalau capek bisa berenti di gazebo di pinggiran ada di bbrp titik ga bawa baju ganti aman banyak yg jual di lokasi ada makanan n minuman juga dan souvenir kita pesen onsen private harga rb buat jam cukup buat orang ada kamar mandi bilasnya tp sayang krn dah agak sore air kamar mandi udh mati nyala hehe tp air panas n view nya worthed banget sewa handuk nya k pr nya siy pas balik ke atas mendaki banget wkwkw tp semuanya seru dan worthed banget	['masuk', 'lewat', 'kebun', 'raya', 'baturaden', 'lewatin', 'bobocabin', 'dll', 'jalan', 'terus', 'tp', 'jalannya', 'lumayan', 'bagus', 'siy', 'pas', 'deket', 'lokasi', 'pintu', 'gerbang', 'jalur', 'pendakian', 'baru', 'mulai', 'gerasak', 'geruduk', 'jalan', 'nya', 'sampe', 'lokasi', 'parkir', 'wow', 'banget', 'hutan', 'bayar', 'seikhlasnya', 'kita', 'pakai', 'motor', 'kasih', 'rb', 'tp', 'pas', 'pulang', 'kang', 'parkir', 'dah', 'gada', 'wkwkw', 'soale', 'dah', 'mulai', 'gelap', 'juga', 'kabut', 'udh', 'turun', 'n', 'sepi', 'buka', 'sampe', 'jam', 'aja', 'dari', 'parkiran', 'masih', 'jalan', 'kaki', 'meteran', 'turun', 'tp', 'jalannya', 'bagus', 'udh', 'bertangga', 'kalau', 'capek', 'bisa', 'berenti', 'di', 'gazebo', 'di', 'pinggiran', 'ada', 'di', 'bbrp', 'titik', 'ga', 'bawa', 'baju', 'ganti', 'aman', 'banyak', 'yg', 'jual', 'di', 'lokasi', 'ada', 'makanan', 'n', 'minuman', 'juga', 'dan', 'souvenir', 'kita', 'pesen', 'onsen', 'private', 'harga', 'rb', 'buat', 'jam', 'cukup', 'buat', 'orang', 'ada', 'kamar', 'mandi', 'bilasnya', 'tp', 'sayang', 'krn', 'dah', 'agak', 'sore', 'air', 'kamar', 'mandi', 'udh', 'mati', 'nyala', 'hehe', 'tp', 'air', 'panas', 'n', 'view', 'nya', 'worthed', 'banget', 'sewa', 'handuk', 'nya', 'k', 'pr', 'nya', 'siy', 'pas', 'balik', 'ke', 'atas', 'mendaki', 'banget', 'wkwkw', 'tp', 'semuanya', 'seru', 'dan', 'worthed', 'banget']
tempatnya asik dingin dan kadang berkabut pemandangan bagus khas pegunungan perlu fisik yg bagus untuk mendaki tangga pulangnya akses jalan menuju lokasi ada yg rusak toilet di pintu masuk kurang memadai	['tempatnya', 'asik', 'dingin', 'dan', 'kadang', 'berkabut', 'pemandangan', 'bagus', 'khas', 'pegunungan', 'perlu', 'fisik', 'yg', 'bagus', 'untuk', 'mendaki', 'tangga', 'pulangnya', 'akses', 'jalan', 'menuju', 'lokasi', 'ada', 'yg', 'rusak', 'toilet', 'di', 'pintu', 'masuk', 'kurang', 'memadai']

After the tokenization stage, we move on to the normalization stage. Normalization is the process of converting data or text into a uniform format. It aims to transform informal terms, such as slang, abbreviations, or irregular spellings, into a standard form that complies with language rules. This process is carried out to make the text more consistent, organized, and easier to analyze by machine learning algorithms and other analysis methods.

Figure 4. Normalization Stage Script

Figure 4 shows the script used in the text normalization process. At this stage, the normalization dictionary is read from an Excel file named "normalisasi_kamus.xlsx" which contains a list of non-standard words and their standard equivalents. The data is then loaded into a DataFrame named kamus_df and converted into a dictionary data structure (*Normalization Dictions*). The dictionary is used as a reference to replace each non-standard word in the tokenization results with its standard form during the text normalization process.

C. Stopword Removal

Stopword removal is a crucial step in text data preprocessing, aiming to eliminate common words with high frequency but low informational value in the analysis context. These words, known as stopwords, commonly appear in everyday conversation and do not significantly influence the interpretation of the overall meaning of the text. In this study, the stopword removal process was carried out using a word list previously compiled by the researcher according to the analysis requirements. In this study, the stopword removal process was performed by combining the default stopword list from NLTK Indonesian with an additional stopword dictionary in Excel. This combination aims to better align the stopword removal process with the characteristics of the review data used. An example of stopword removal is shown in Table 3.

Table 3. Stopword Removal Result Example

Before	After
['masuk', 'lewat', 'kebun', 'raya', 'baturraden', 'melewati', 'bobocabin', 'dll', 'jalan', 'terus', 'tapi', 'jalan', 'lumayan', 'bagus', 'sih', 'pas', 'dekat', 'lokasi', 'pintu', 'gerbang', 'jalur', 'pendakian', 'baru', 'mulai', 'berisik', 'geruduk', 'jalan', 'nya', 'sampai', 'lokasi', 'parkir', 'wow', 'banget', 'hutan', 'bayar', 'seikhlasnya', 'kita', 'pakai', 'motor', 'kasih', 'ribu', 'tapi', 'pas', 'pulang', 'abang', 'parkir', 'sudah ', 'tidak ', 'wkwkw', 'soale', 'sudah ', 'mulai', 'gelap', 'juga', 'kabut', 'sudah', 'turun', 'dan', 'sepi', 'buka', 'sampai', 'jam', 'saja', 'dari', 'parkiran', 'masih', 'jalan', 'kaki', 'meter', 'turun', 'tapi', 'jalan', 'bagus', 'sudah', 'bertangga', 'kalau', 'capek', 'bisa', 'berenti', 'di', 'gazebo', 'di', 'pinggiran', 'ada', 'di', 'beberapa', 'titik', 'tidak', 'bawa', 'baju', 'ganti', 'aman', 'banyak', 'yang', 'jual', 'di', 'lokasi', 'ada', 'makanan', 'dan', 'minuman', 'juga', 'dan', 'souvenir', 'kita', 'pesen', 'pemandian air panas', 'private', 'harga', 'ribu', 'buat', 'jam', 'cukup', 'buat', 'orang', 'ada', 'kamar', 'mandi', 'bilasnya', 'tapi', 'sayang', 'karena', 'sudah ', 'agak', 'sore', 'air', 'kamar', 'mandi', 'sudah', 'mati', 'nyala', 'hehe', 'tapi', 'air', 'panas', 'dan', 'pemandangan', 'nya', 'layak', 'banget', 'sewa', 'handuk', 'nya', 'ke', 'susah', 'nya', 'sih', 'pas', 'balik', 'ke', 'atas', 'mendaki', 'banget', 'wkwkw', 'tapi', 'semuanya', 'seru', 'dan', 'layak', 'banget']	['masuk', 'kebun', 'raya', 'baturraden', 'melewati', 'bobocabin', 'jalan', 'jalan', 'lumayan', 'bagus', 'lokasi', 'pintu', 'gerbang', 'jalur', 'pendakian', 'berisik', 'geruduk', 'jalan', 'lokasi', 'parkir', 'hutan', 'bayar', 'seikhlasnya', 'motor', 'kasih', 'pulang', 'abang', 'parkir', 'sudah ', 'tidak ', 'soale', 'sudah ', 'gelap', 'kabut', 'turun', 'sepi', 'buka', 'parkiran', 'jalan', 'kaki', 'meter', 'turun', 'jalan', 'bagus', 'bertangga', 'capek', 'berenti', 'gazebo', 'pinggiran', 'titik', 'bawa', 'baju', 'ganti', 'aman', 'jual', 'lokasi', 'makanan', 'minuman', 'souvenir', 'pesen', 'pemandian air panas', 'private', 'harga', 'orang', 'kamar', 'mandi', 'bilasnya', 'sudah ', 'sore', 'air', 'kamar', 'mandi', 'mati', 'nyala', 'air', 'panas', 'pemandangan', 'layak', 'sewa', 'handuk', 'susah', 'mendaki', 'seru', 'layak']
['tempat', 'asyik', 'dingin', 'dan', 'kadang-kadang', 'berkabut', 'pemandangan', 'bagus', 'khas', 'pegunungan', 'perlu', 'fisik', 'yang', 'bagus', 'untuk', 'mendaki', 'tangga', 'pulangnya', 'akses', 'jalan', 'menuju', 'lokasi', 'ada', 'yang', 'rusak', 'toilet', 'di', 'pintu', 'masuk', 'kurang', 'memadai']	['asyik', 'dingin', 'kadang-kadang', 'berkabut', 'pemandangan', 'bagus', 'khas', 'pegunungan', 'fisik', 'bagus', 'mendaki', 'tangga', 'pulangnya', 'akses', 'jalan', 'lokasi', 'rusak', 'toilet', 'pintu', 'masuk', 'memadai']

D. Labeling & Splitting Data

After data preprocessing is complete and all data has been successfully processed, the next step is the labeling process, which aims to classify the data into sentiment categories, namely positive and negative. This labeling is carried out using a lexicon-based approach using a dictionary of positive and negative Indonesian words obtained from the ID-Opinion Words repository. The labeling results can be seen in Figure 5. Based on the graph, the number of reviews with positive sentiment is greater than the number of negative reviews. A total of 1,708 positive reviews and 1,204 negative reviews were recorded. These results indicate that the majority of reviews given by visitors tend to be positive.

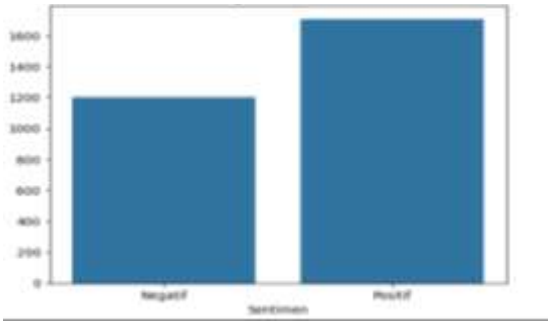


Figure 5. Labeling Chart

Data division, also known as data splitting, in the training process involves separating the data into two parts: training data and test data. To find the best data split composition, several experiments were conducted to evaluate various data split scenarios. In this study, the data splits used were 60:40, 70:30, 80:20, and 90:10. The results of each experiment can be seen in the following table.

Table 4. Results of the data split process

Phase	Training	Testing	Accuracy
1	60	40	90%
2	70	30	91%
3	80	20	92%
4	90	10	95%

From the data in Table 4 it can be seen that a 90:10 data split, with 90% used as training data and 10% for testing, yielded the best results, with an accuracy rate of 95%. Therefore, in the next stage, a 90:10 data ratio will be used in the dataset.

TF-IDF is used in text processing to assign weights to each word based on its frequency in a document and how rarely it appears across the entire document. Therefore, more informative words will have greater weight.

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
```

Figure 6. Install the TF-IDF Library

The pandas library is used for data manipulation and analysis. Using pandas, data from various sources such as Excel files, CSV files, or databases can be read and managed in a structure called a DataFrame, facilitating further analysis. Next, import the TfidfVectorizer class from the sklearn.feature_extraction module. This class is used to convert a text data set into a numeric representation based on TF-IDF (Term Frequency-Inverse Document Frequency) calculations.

```
#TF-IDF vectorizer
vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(dataanalisis['processed_text'])
```

Figure 7. TF-IDF Script

The script in Figure 4.14 above is used to convert text data into a numeric representation using the TF-IDF method. The Tfidf-Vectorizer() function from the scikit-learn library is used to create a vectorizer object. This object functions to convert a collection of text into a number by calculating the weight of each word based on the TF-IDF method. Next, the transform (data-analisys['processed_text'] command is run to apply the TF-IDF process to the data contained in the 'processed_text' column of the Data-Frame. At this stage, the system automatically learns all existing words, then calculates the TF-IDF value for each word in each document. The results of this process are stored in a variable called tfidf matrix, which is a numeric matrix that represents the weight of each word.

E. SMOTE Balancing

SMOTE (Synthetic Minority Oversampling Technique) is an oversampling technique used to balance the amount of data in each class by creating synthetic samples (new data) for the minority class. This technique is applied after the text data is converted to numeric form, so the model is not biased towards the majority class during training. The SMOTE object is initialized with the parameter random_state=42. This parameter ensures that the synthetic data generation process remains consistent each time the program is run. Using the fit_resample function, the SMOTE method first learns the data distribution patterns in the minority class based on the TF-IDF extraction results (x_train_tfidf). Next, SMOTE generates new synthetic data for the minority class until the data balance between the classes is achieved. The result of this process is new training data (x_train_smote) and new training data labels (y_train_smote), which are balanced and ready to be used in the classification model training stage.

```
from collections import Counter

print("Sebelum SMOTE:", Counter(y_train))
print("Sesudah SMOTE:", Counter(y_train_smote))

Sebelum SMOTE: Counter({1: 1537, 0: 1083})
Sesudah SMOTE: Counter({0: 1537, 1: 1537})
```

Figure 8. Before and After SMOTE

Figure 8 above shows a comparison of the number of data points for each class before and after applying the SMOTE method. Before SMOTE, the training data was unbalanced, with 1537 data points for class 1 and 1083 for class 0. After SMOTE was applied, the number of data points for both classes was balanced, with 1537 data points for each class. This aims to reduce model bias toward the majority class and improve classification performance.

F. *SVM & XGBoost Analyze*

The Linear Support Vector Classification (LinearSVC) model is used to create the model. The random_state=42 parameter is used to ensure that the model training process produces consistent results every time the program is run. To train the SVM model, use 'svm_model.fit(x_train_smote, y_train_smote)' using training data that has gone through a class imbalance process with the SMOTE method.

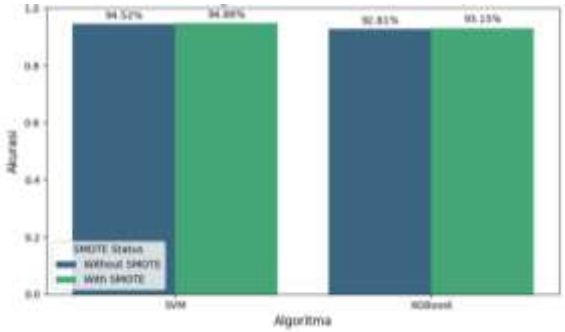


Figure 9. Comparison results of SVM and XGBoost accuracy

Based on Figure 9 above, the test results show the performance of the Support Vector Machine algorithm with an accuracy of 94.52% without SMOTE and 94.86% with SMOTE, which is higher than the XGBoost algorithm with an accuracy of 92.80% without SMOTE and 93.15% with SMOTE. This shows that the consistent accuracy difference of the Support Vector Machine algorithm has a better performance in classifying visitor reviews than XGBoost. This difference in performance may be influenced by the characteristics of text data represented using TF-IDF, which produces high feature dimensions and is sparse. In these conditions, SVM is known to be more effective in finding separating hyperplanes between classes than boosting methods such as XGBoost, so SVM is more optimal than XGBoost. This may be because the boosting method is more susceptible to data distribution classes, so although SMOTE helps improve the representation of minority classes, the resulting increase in accuracy is not very significant. The application of SMOTE to both algorithms resulted in improvements, indicating that before SMOTE, both models performed quite well in classification, even though the data contained class imbalance. Table 5 presents detailed accuracy values for the Support Vector Machine and XGBoost algorithms before and after SMOTE application.

Table 5. SVM and XGBoost Performance Results

Model	*Accuracy*	Desc.
SVM	94,52%	*Without* SMOTE
SVM	94,86%	*With* SMOTE
XGBoost	92,80%	*Without* SMOTE
XGboost	93,15%	*With* SMOTE

The results of this study were then compared with several previous studies. Research conducted by several researchers [5][8][20] showed that the SVM model obtained the highest accuracy of 89.9% higher than KNN and Naïve Bayes. The results of this study are in accordance with that study, because the SVM model also showed the best performance compared to XGBoost with an accuracy of 94.86% after applying SMOTE. The difference in accuracy values shows that the application of SMOTE is able to improve the ability of the SVM model in handling imbalanced data, so that model evaluation is more accurate and represents all classes. This is different from previous studies that have not considered class balance, thus potentially causing bias towards the majority class. Then, research conducted by [2][4][12][16] the K-Nearest Neighbor (KNN) algorithm has superior performance compared to SVM in sentiment classification. This is indicated by a higher accuracy value compared to other algorithms in previous studies. However, different results were obtained in this study, where the SVM algorithm showed higher performance than the XGBoost algorithm. These differences in results indicate that the performance of the classification algorithm is not absolute, but is greatly influenced by the characteristics of the dataset, the amount of data, text preprocessing techniques, feature extraction methods, and the testing flow applied.

CONCLUTIONS

Based on the results of a sentiment analysis conducted on 4,096 visitor reviews of Wana Wisata Baturraden on Google Maps, it can be concluded that the Support Vector Machine algorithm outperforms XGBoost in sentiment classification. The superiority of the Support Vector Machine algorithm is evident in its higher accuracy compared to XGBoost.

The results of this study indicate that the Support Vector Machine algorithm achieved an accuracy of 94.52% without SMOTE, increasing to 94.86% with SMOTE, compared to the XGBoost algorithm, which achieved an accuracy of 92.80% without SMOTE and 93.15% with SMOTE. This indicates that the Support Vector Machine algorithm is more suitable for text-based sentiment classification represented using TF-IDF and has high feature dimensions and sparse data characteristics. While the decision tree-based XGBoost algorithm tends to be less optimal for text data with many zero values, resulting in relatively lower performance in the context of tourism review sentiment analysis.

Acknowlegdement

My gratitude goes to my thesis supervisor, Dr. Irfan Santiko, S.Kom., M.Kom. Thank you for your valuable guidance, criticism, direction, input, and intellectual encouragement, as well as for spending a lot of time patiently and understandingly in assisting me in the process of analyzing my data. Lastly, my friends and all parties who have provided moral support, assistance, and motivation to me during the preparation of this research, whom I cannot mention one by one.

Author Contributions

C.R., Analysis, Writing, Data Collecting, I.S. Analisys Written.

Funding

Not applicable.

REFERENCE

[1] N. K. Diwangkara, S. R. Sari, and R. S. Rukayah, "Pengembangan Pariwisata Kawasan Baturraden," J. Arsit. ARCADE, vol. 4, no. 2, p. 120, 2020, doi: 10.31848/arcade.v4i2.431.
[2] B. Ramadhani and R. R. Suryono, "Komparasi Algoritma Naïve Bayes dan Logistic Regression Untuk Analisis Sentimen Metaverse," J. Media Inform. Budidarma, vol. 8, no. 2, p. 714, 2024, doi: 10.30865/mib.v8i2.7458.
[3] I. S. Djunaid, "Penyuluhan Pentingnya Pemahaman Siswa SMK Pariwisata Tentang Skill Yang Dibutuhkan Dalam Dunia Kerja Pariwisata Di SMK Darmawan Bogor," vol. 5, no. 1, pp. 36–46, 2021.
[4] M. E. Apriyanti, H. Subiyantoro, and P. Astuti, "Pengaruh Sektor Pariwisata Terhadap Pendapatan Asli Daerah Dan Dampaknya Pada Penyerapan Tenaga Kerja Di Setiap Kabupaten Provinsi Bali Tahun 2019," JABE (Journal Appl. Bus. Econ., vol. 9, no. 4, p. 462, 2023, doi: 10.30998/jabe.v9i4.14995.
[5] W. Widyaningsih, E. Nurwati, and S. D. Nugroho, "The Effect of e-WOM and Destination Image on Tourist Loyalty through Tourist Satisfaction," J. Ilm. MEA, vol. 4, no. 1, pp. 522–540, 2020.
[6] I. Khotimah and R. Sulistyowati, "Pengaruh Electronic Word Of Mouth (EWOM) di Media Sosial terhadap Minat dan Keputusan Berkunjung Di Surabaya," J. Pendidik. Tata Niaga, vol. 10, no. 1, pp. 1679–1688, 2022.
[7] W. Khofifah, D. N. Rahayu, and A. M. Yusuf, "Analisis Sentimen Menggunakan Naive Bayes Untuk Melihat Review Masyarakat Terhadap Tempat Wisata Pantai Di Kabupaten Karawang Pada Ulasan Google Maps," J. Interkom J. Publ. Ilm. Bid. Teknol. Inf. dan Komun., vol. 16, no. 4, pp. 28–38, 2022, doi: 10.35969/interkom.v16i4.192.
[8] M. D. Setiawan and S. Mariska, "Analisis Sentimen Ulasan Pengunjung Sirkuit Mandalika Pada Google Mpas Dengan Metode Naive Bayes," vol. 5, pp. 955–965, 2025.
[9] W. Atmojo, V. Atina, and H. Permatasari, "Analisis Sentimen Pelanggan Pada Ulasan Google Maps Restoran Al-Ghiff Steak Menggunakan Model Indobert," Simtek J. Sist. Inf. dan Tek. Komput., vol. 10, no. 2, pp. 336–343, 2025, doi: 10.51876/simtek.v10i2.1602.
[10] A.S. Yondra, D. Triyanto, and S. Bahri, "Implementasi Web Scraping Untuk Mengumpulkan Informasi Produk Dari Situs E-Commerce Dan Marketplace Dengan Teknik Pemrosesan Paralel," Coding J. Komput. dan Apl., vol. 10, no. 01, p. 93, 2022, doi: 10.26418/coding.v10i01.52722.

[11] S. Satriajati, S. B. Panuntun, and S. Pramana, "Implementasi Web Scraping Dalam Pengumpulan Berita Kriminal Pada Masa Pandemi Covid-19," Semin. Nas. Off. Stat., vol. 2020, no. 1, pp. 300–308, 2021, doi: 10.34123/semnasoffstat.v2020i1.578.
[12] A. Rahmatulloh and R. Gunawan, "Web Scraping with HTML DOM Method for Data Collection of Scientific Articles from Google Scholar," Indones. J. Inf. Syst., vol. 2, no. 2, pp. 95–104, 2020, doi: 10.24002/ijis.v2i2.3029.
[13] T. P. Sihaloho, D. E. Ratnawati, and B. Rahayudi, "Analisis Sentimen Objek Wisata Danau Toba berdasarkan Ulasan Pengunjung menggunakan Algoritma Support Vector Machine," J. Pengemb. Teknol. Inf. dan Ilmu Komput., vol. 6, no. 9, pp. 4204–4209, 2022.
[14] N. Utami and A. Made, "Text Mining Dalam Analisis Sentimen Pembelajaran Daring Di Masa Pandemi Covid 19 Menggunakan Algoritma K-Nearest Neighbor," vol. 4, no. 1, pp. 59–64, 2022.
[15] T. Ridwansyah, "Implementasi Text Mining Terhadap Analisis Sentimen Masyarakat Dunia Di Twitter Terhadap Kota Medan Menggunakan K-Fold Cross Validation Dan Naïve Bayes Classifier," KLIK Kaji. Ilm. Inform. dan Komput., vol. 2, no. 5, pp. 178–185, 2022, doi: 10.30865/klik.v2i5.362.
[16] M. P. R. Putra and K. R. N. Wardani, "Penerapan Text Mining Dalam Menganalisis Kepribadian Pengguna Media Sosial," JUTIM (Jurnal Tek. Inform. Musirawas), vol. 5, no. 1, pp. 63–71, 2020, doi: 10.32767/jutim.v5i1.791.
[17] L. Pertiwi, "Penerapan Algoritma Text Mining, Steaming Dan Texrank Dalam Peringkasan Bahasa Inggris," BIMASATI (Bulletin Multi-Disciplinary Sci. Appl. Technol., vol. 1, no. 3, pp. 100–104, 2022.
[18] O. H. Rahman, G. Abdillah, and A. Komarudin, "Classification of Hate Speech on Social Media Twitter Using Support Vector Machine," RESTI J. (Systems Eng. Inf. Technol., vol. 5, no. 1, pp. 17–23, 2021.
[19] M. U. Albab, Y. Karuniawati, and M. N. Fawaiq, "Optimization of the Stemming Technique on Text Preprocessing President 3 Periods Topic," J. Transform., vol. 20, no. 2, pp. 1–12, 2023, doi: 10.26623/transformatika.v20i2.5374.
[20] R. Wati, S. Ernawati, and H. Rachmi, "TF-IDF Weighting Using Naïve Bayes on Public Sentiment on The Issue of Rising BIPIH," J. Manaj. Inform., vol. 13, no. April, pp. 84–93, 2023.
[21] I. W. B. Suryawan, N. W. Utami, and K. Q. Fredlina, "Analisis Sentimen Review Wisatawan pada Objek Wisata Ubud Menggunakan Algoritma Support Vector Machine," J. Inform. Teknol. dan Sains, vol. 5, no. 1, pp. 133–140, 2023.
[22] E. R. N. Mustaqim, U. Pagalay, and C. Crysdian, "Prediksi Tingkat Kepercayaan Masyarakat Terhadap Pilpres 2024 Menggunakan Tf-Idf Dan Bow Menggunakan Metode Svm," Mandalika ISSN 2721 …, pp. 515–530, 2024.