KLIK: Kajian Ilmiah Informatika dan Komputer
ISSN 2723-3898 (Media Online)
Vol 4, No 6, June 2024, Hal 2763-2776
DOI 10.30865/klik.v4i6.1889
https://djournals.com/klik

Implementation of Aspect-Based Sentiment Analysis on the Mitra Darat App User Reviews Using Machine Learning

Maulaya Ishaq*, Dewi Lestari, Michael Marchenko

Informatics Engineering, Universitas Trilogi, Jakarta, Indonesia
Email: [1] maulaya_ishaq@trilogi.ac.id, [2] dewy24@trilogi.ac.id, [3,] michael@universitas-trilogi.ac.id
Email Corresponding Author: maulaya_ishaq@trilogi.ac.id

Abstract−The Mitra Darat application is an Android-based application available on the Google Play Store, developed by the Directorate General of Land Transportation (Ministry of Transportation). User reviews on this platform enable direct communication with developers, offering valuable feedback for service enhancement and future development. For that reason, aspect-based sentiment analysis is needed to help organizations monitor product sentiment in user feedback, and understand user needs. This research aims to implement aspect-based sentiment analysis on Mitra Darat application user reviews to generate insights via a system dashboard. Comparing Naive Bayes (NB) and Support Vector Machines (SVM) for machine learning models with the addition of pre-trained Indobert as word embedding, SVM showed superior performance with an accuracy score of 94% for aspect classification and 90% for sentiment classification, compared to Naive Bayes with scores of 84% and 78% respectively. The trained Support Vector Machine model (SVM) was then utilized to analyze 967 reviews of the Mitra Darat application for 2023. The results of the analysis are presented on a dashboard page with summary information, which shows that the overall user sentiment is 52.3% positive, 10% neutral, and 37.7% negative. In terms of sentiment polarity by aspect, the system aspect is 29% positive, 8% neutral, and 63% negative, meaning that some bugs and issues have been found in the application, so it can be evaluated for future system development. The service aspect is 62% positive, 27% neutral, and 11% negative, which means that the free mudik service is quite well organized.

Keywords: Aspect-based Sentiment Analysis; Machine Learning; Support Vector Machine; Naive Bayes; Mitra Darat.

1. INTRODUCTION

In early 2023, the Directorate General of Land Transportation of the Ministry of Transportation launched the *Mitra Darat* application to digitally transform public services in the land transportation sector. *Mitra Darat* is a multi-service application that provides various information related to the supervision, licensing, and operations of land transportation on a single platform [1]. The application's primary service is being the platform for free *mudik* registration. *Mudik* is the activity of migrant workers to return to their hometowns. *Mudik* in Indonesia is synonymous with an annual tradition that takes place before major religious holidays such as Eid al-Fitr, Eid al-Adha, Christmas & New Year [2]. The *Mitra Darat* App is distributed through the Google Play store and has been downloaded by more than 100,000 people, and reviewed by more than 1000 users.

User reviews on the Google Play Store are a direct communication channel between users and application owners or developers [3]. Thus, user reviews are one of the most critical indicators for evaluating the application's service and making appropriate recommendations for improving the application's functionality [4]. The problem is that the application owner has not utilized this review data as evaluation material. The Directorate General of Land Transportation also has not visualized data that illustrates user perspectives regarding the reliability of the application and the service provided.

Sentiment analysis is a possible solution for the problems mentioned above. Many companies use sentiment analysis to analyze customer feedback on social media platforms, review sites, and other channels [5]. Sentiment analysis or opinion mining is typically used to analyze textual data and extract opinions and sentiments from the data using various natural language processing techniques [6]. The results can be positive, negative, and neutral. Further development towards sentiment analysis is Aspect-Based Sentiment Analysis (ABSA). Aspect-based methods allow organizations to extract the most important aspects from customer feedback and services [7]. By applying aspect-based sentiment analysis, organizations can monitor the sentiment of user reviews based on specific aspects such as systems and services, enabling real-time understanding of user needs and discovering issues for future evaluation and development.

In the last decade, there have been many studies related to the sentiment analysis of user reviews using various techniques. Such as research conducted by Nur Hakim et al [8] on Indihome user reviews taken from Google Play between November 1st , 2020 - December 15th , 2020 with a total of 2539 reviews. Researchers evaluated two classification models between SVM and Naive Bayes with three scenarios of splitting the data between training data and testing data, 70: 30, 80: 20 and 90: 10. In each scenario, five trials were conducted and the results showed that the average accuracy value of SVM was 86.54% and Naive Bayes 84.69%. In another study by Indrayuni et al [9] using user reviews data from the Halodoc application on Google Play, the researchers compared three machine learning models Naive Bayes, SVM and KNN. From the model testing results, the accuracy value of Naive Bayes is 92.5%, SVM is 93% and KNN is 95%. Further research in 2022 by Nurthohari [10] on public comments on Twitter related to Jakarta Bus Rapid Transportation Services using SVM and TF IDF as feature extraction. His results showed that the SVM method with Linear kernel, four coefficients, and 4000 features outperformed all test models in predicting sentiment with an average accuracy of 92.00%, precision of 91.00%, recall of 92.00%, and support of 2123.

In addition, ABSA research on user reviews including those conducted by CA Bahri et al [11] on 1890 Google Maps user reviews on Bromo Tengger Semeru National Park tourist destinations. The classification models used are

machine learning and transfer learning. Based on the results of his research on the machine learning model, SVM provides a better performance value compared to the Naive Bayes and Linear Regression models with an accuracy value of 89.16% and F1 Score of 62.23%. The next research related to ABSA was conducted by Mubarok et al [12] using the Naive Bayes method with model performance for aspect classification producing F1-Measure of 88.13%, and for sentiment classification producing F1-Measure of 75%.

SVM and Naive Bayes algorithms were chosen because, based on many previous studies, these two methods are the most frequently used methods for sentiment analysis. They produce optimal accuracy and are suitable for handling small data training. This research compares the Support Vector Machine (SVM) and Naive Bayes algorithms to determine which model performs better in classifying sentiment and aspects in The Mitra Darat Application user reviews. SVM has been tested for text classification by Joachims (1998) and produced good performance in all experiments with a lower error rate than other classification methods [13]. Meanwhile, Naive Bayes is a collection of classification algorithms based on Bayes' Theorem; these algorithms provide excellent results when used for text data analysis [14].

As a comparison to see the effectiveness of using machine learning methods (SVM and Naïve Bayes) in handling aspect-based sentiment analysis tasks on user reviews, a different method has been carried out by Dwi Intan Af'idah et al [15] using deep learning (LSTM and Bi-LSTM). The study focused on visitor reviews of 10 priority tourist destinations in Indonesia. The results indicated that Bi-LSTM outperformed LSTM in aspect and sentiment classification, achieving an average accuracy and f1 score of 92.22% and 71.06% compared to LSTM at 90.63% and 70.42%.

In this research, after finding the best machine learning model, it is then implemented into a web-based system dashboard prototype using the Flask framework. Flask is a micro-framework designed for rapid web application development. Flask implements only core functionality allowing developers to add functionality as needed during implementation [16].

Based on the above description and previous studies, this research differs from previous studies because it utilizes a pre-trained Indobert Model for word embedding and implements a web-based dashboard to present the results of sentiment analysis based on user reviews of the *Mitra Darat* application. This research aims to extract data from user reviews of the *Mitra Darat* application and present the results of the analysis through a dashboard display. This will help organizations (Directorate General of Land Transportation of the Ministry of Transportation) monitor sentiment related to system and service aspects in user reviews, and understand user needs for the *Mitra Darat* application, especially Free *Mudik* Service.

2. RESEARCH METHODOLOGY

This research used an aspect-based sentiment analysis method using machine learning as a classification model. The researchers compared two machine learning models: Naive Bayes (NB) and Support Vector Machine (SVM). The model with the best performance value was selected to be implemented into the prototype dashboard system. The following stages were conducted in this research as shown in Figure. 1.

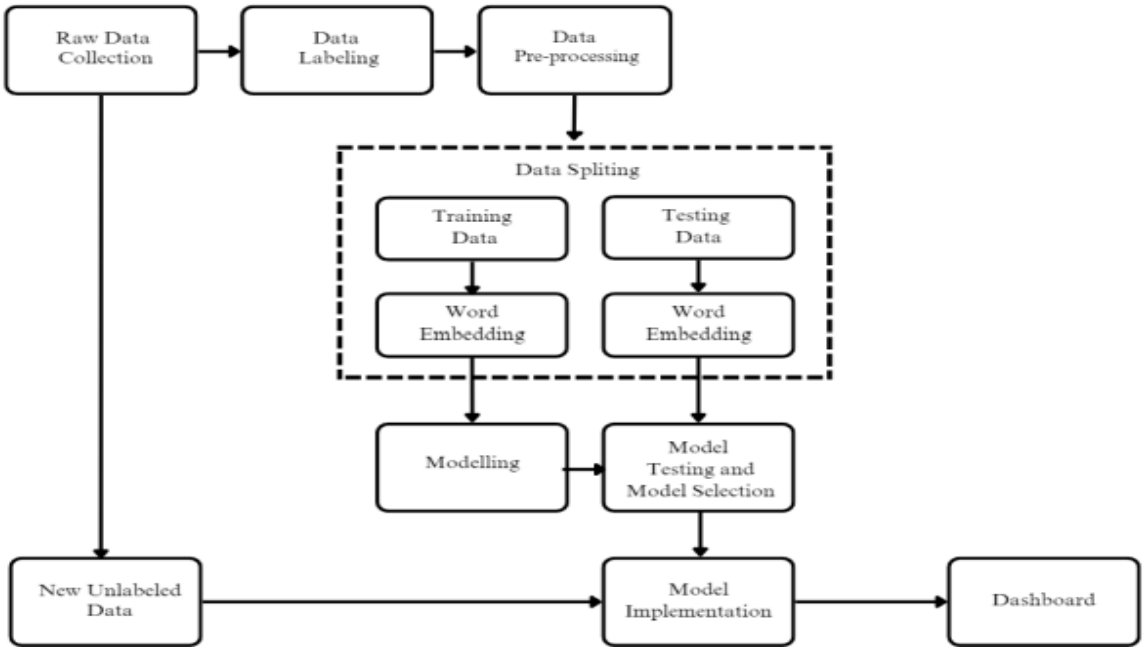


Figure 1. Research Procedure

Figure. 1 describes the stages of the research, starting from data collection, data labeling, data pre-processing, data splitting, word embedding, modeling, model selection, and finally implementing the model into the prototype dashboard system.

2.1 Data Collection

Data from user reviews of the *Mitra Darat* application on Google Play was retrieved using the Google Play Scraper library in Python. The data collection phase was conducted in November 2023 from all available review data from January to November 2023. The output is a CSV (Comma Separated Values) file with 980 data records in the form of review sentences.

2.2 Data Labeling

In supervised machine learning, data labeling or annotation is required to train the model. The process of labelling the sentiment polarity of each review uses a crowd sourcing method where the annotators come from the general public who are not linguistic experts. To reduce subjectivity bias, there are three annotators. Each annotator labeled the reviews with positive, negative and neutral polarity values. The data annotation principles for the annotators follow the general principle that positive labels are given to sentences that have evaluative content such as good, useful, helpful, etc. Negative labels are given for sentences with complaint content such as difficult, not good, slow, error, etc. Neutral labels are given to annotation that contain suggestions, questions and reviews caused by ticket exhaustion. Annotation results are taken from the most labels on each review. If there are differences in labels for each annotator, the label that matches the rating is taken, where a rating above 4 is positive, 3 is neutral, and below 2 is negative. For aspect annotation, the words bag rule is used, as shown in Table 1 Examples of sentiment and aspect annotation results as shown in Table 2.

Table 1. Bag of words for aspect labeling

Aspect	Key words
System/Apps	'http failed', 'failed', 'error', 'eror', *'tidak bisa login'*, *'lemot'*, 'down', *'gagal login'*, 'login', *'otp'*, *'nomer'*, *'no'*, 'menu', 'log in', *'kendala'*, *'fitur'*, *'update versi'*, *'aplikasi'*, *'apk'*, *'loading'*, *'apl'*, *'tidak kompatibel'*, *'tidak bisa masuk'*, *'akun'*, *'email'*, *'imail'*, *'lambat'*, *'data'*, *'http'*, *'tidak bisa akses'*, *'ngecek'*, *'apps'*, *'nge lag'*, 'download', *'fungsi'*, *'jaringan'*, *'donwload'*, *'app'*,*'unduh'*, *'masuk susah'*, *'tidak bisa'*, *'daftar'*, 'try again', *'update'*
Service (Free *Mudik*)	*'pelayanan'*, *'layanan'*, *'mudik gratis'*, *'mudik'*, *'tiket'*, *'motis'*, *'pendaftaran'*, *'daftar'*, *'tujuan'*, *'verifikasi'*, *'wilayah'*, *'berangkat'*, *'kota'*, *'armada'*, *'terminal'*, *'kloternya'*, *'gratis'*, *'kuota habis'*, *'kuota'*, *'abis'*, *'pulang kampung'*, *'bus'*, *'bis'*, *'akap'*
Others	exclusion of the above key words

Table 1 shows the keywords used for aspect labeling in user reviews of The *Mitra Darat* application.

Table 2. Data user review with bahasa indonesia labels

Reviews	**Sentiment**	**Aspects**
Tidak bisa login tulisanya failed coba ganti akun akun tulis failed	Negative	System
Ga jelas petugas bilang jam 10 berangkat, malah jam 8 berangkatnya. Tolong bilangin petugasnya kasih petunjuk yang benar.	Negative	Service (Free *Mudik*)
Alhamdulillah sangat membantu dalam perjalanan mudik gratis saya thun ini trimksh Mitra Darat	Positive	Service (Free *Mudik*)
Aplikasi bagus semoga bermanfaat	Positive	System
Mantap	Positive	Others

Table 2 presents examples of sentiments and aspects labeled for the *Mitra Darat* application based on the data labeling process.

2.3 Data Pre-processing

Data pre-processing is needed to remove noise, symbols, or punctuation from the reviews in order to obtain clean data prior to by machine learning processing. The pre-processing stages performed in this research are case folding, cleansing and tokenization, normalization, negation handling, and stop-word removal as seen in in Figure. 2.

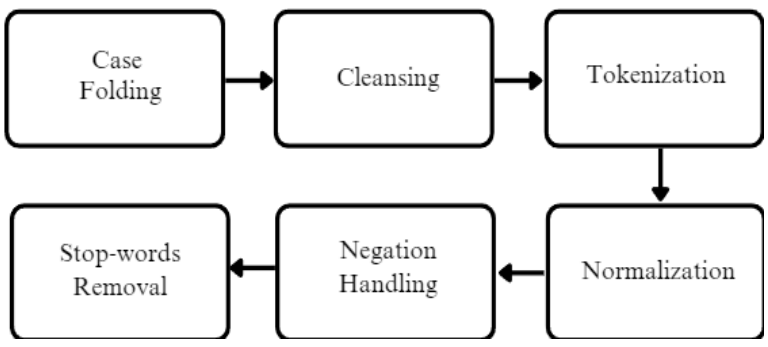


Figure 2. Data Pre-processing Stages

2.3.1 Case Folding

The process of converting all letters into lowercase or uppercase letters. This technique projects all words in the text into the same feature space [17].

2.3.2 Cleansing

A process that removes unimportant texts such as (@, #, link), numeric characters, non-alphabetic characters, punctuation marks, empty characters, emoticon/emoji characters, and repetitive characters to make the training process more efficient and easier.

2.3.3 Tokenization

The process of separating sentences into words, phrases, or other meaningful parts, called tokens [18].

2.3.4 Normalization

The process of converting informal words into formal words according to KBBI standards because the data obtained from scraping the user comments is mainly filled with foreign or informal words

2.3.5 Negation Handling

The process of handling the negation of a word. It will combine the words that come after the negation word marked with a separator sign (_). However, the negation word remains and does not disappear because it will be removed during the Stop-word process, leaving only the word resulting from the negation handling.

2.3.6 Stop-words Removal

The process by which unimportant high-frequency words are removed. Examples of stop-words are the conjunctions "and", "or," "but," "will," and others. This stop-words removal uses a list of Indonesian stop-words derived from the NLTK (Natural Language Tool Kit) module.

2.4 Data Splitting

Data splitting is commonly used in machine learning to divide data into training and test data. In this research, 80% of the data is used as training data, while the remaining 20% is used as test data.

2.5 Word Embedding

Word Embedding represents words as vectors generated by examining large data sets, where words with the same meaning (semantically similar) will have adjacent vectors in the vector space [19]. Vectorized word embedding, neural networks are utilized to produce high-quality embedding. A pre-trained Indobert model was used in this research because it has been trained on a large scale on an Indonesian language corpus consisting of about 4 billion words [20].

2.6 Modelling

Machine Learning models are used in sentiment analysis to learn how to labeling sentiment on new data based on annotated training data [21] or supervised learning methods. In this study, the researchers compared two widely used machine learning algorithms in sentiment analysis namely Naive Bayes and Support Vector Machine.

The Naive Bayes method is a set of supervised learning algorithms based on the application of Bayes' theorem with "naive" assumptions about the conditional independence between each pair of features given a class variable value [22]. To calculate the probabilities with Bayes' Theorem by using the equation (1).

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)} \quad (1)$$

In this equation, basically we are trying to find probability of event A, given the event B is true. Event B is also termed as evidence. Where P(B|A) is the probability of B given that A is true, while P(A) and P(B) are the independent probabilities of A and B.

In this research, the Naive Bayes method was used with the Gaussian Naive Bayes type. The probability of the feature is assumed to be Gaussian and its values are obtained by the equation (2).

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) \quad (2)$$

The Gaussian Naive Bayes algorithm assumes that the features in the dataset are normally (Gaussian) distributed. This means that the values of each feature follow a bell-shaped curve when plotted on a graph. When classifying a new data point x, the algorithm identifies the maximum value of the posterior probability of each class and assigns the data point to that class. The normal distribution is defined by two parameters: the mean (μ) and the standard deviation (σ). The mean indicates the central value of the distribution, while the standard deviation measures the spread of the values around the mean. Assuming a normal distribution simplifies the calculation of probabilities in the Naive Bayes algorithm. Rather

than attempting to estimate the full probability distribution of each feature, we only need to estimate the mean and standard deviation. This approach makes the algorithm computationally efficient and easy to implement.

Support Vector Machines (SVM) are a set of supervised machine learning methods that create a boundary separating two groups based on a set of provided training data [23]. The pattern is stored in a model that aims to predict the labeling used in the testing phase. SVM finds the maximum separating hyperplane (the hyperplane with the maximum distance between the closest training tuples). The main idea behind SVM in classification tasks is to get a good separation between classes [24], as shown in the Figure. 3.

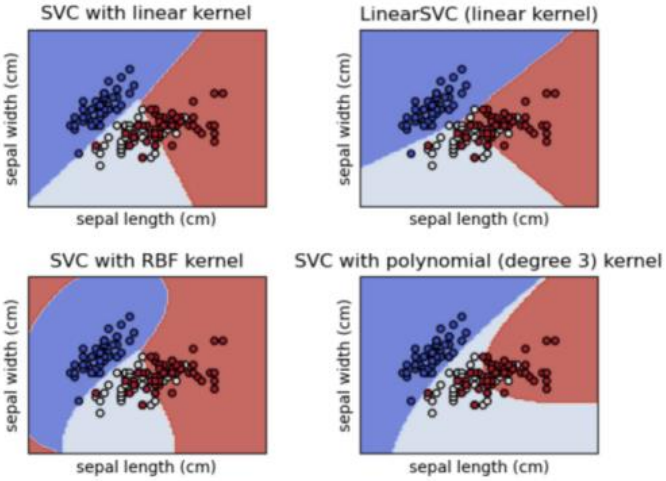


Figure. 3 Hyperplane in SVM

Figure. 3 illustrates the hyperplane with different SVM kernels, including SVM with a linear kernel, SVM with an RBF kernel, and SVM with a polynomial kernel. The utilization of kernel tricks can greatly enhance the performance of the SVM algorithm when addressing various classification tasks.

2.7 Model Selection

Model evaluation was performed to determine which model, produces the best performance between Naive Bayes and Support Vector Machine. The evaluation method used was the confusion matrix (Figure. 4).

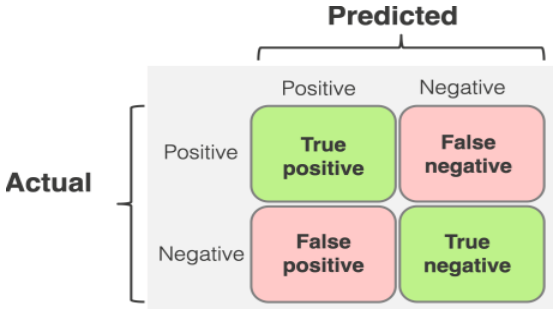


Figure. 4 Confusion Matrix

The Confusion Matrix in Figure. 4 displays the number of accurate predictions (True Positive and True Negative) and the number of inaccurate predictions (False Positive and False Negative). By determining the True Positive, True Negative, False Positive, and False Negative values, we can calculate various metrics to evaluate the model's performance, including accuracy, precision, recall, and f-1 score using the equations below.

2.7.1 Accuracy

Measures the extent to which the model can classify correctly and can be calculated by using the equation (3).

$$\text{Accuracy} = \frac{\text{True Postive + True Negative}}{\text{TP+FP+TN+FN}} \qquad (3)$$

The accuracy metric is calculated by dividing all correct predictions (True Positive + True Negative) by the total number of predictions (True Positive + False Positive + True Negative + False Negative).

2.7.2 Precision

Measures the extent to which the positive outcomes predicted by the model are actually positive. The precision is assessed using equation (4).

$$\text{Precision} = \frac{\text{True Postive}}{\text{True Positive} + \text{False Positive}} \quad (4)$$

The precision metric is calculated by dividing the correctly identified positive results (True Positive) by the total number of positive predictions (True Positive + False Positive) made by the model.

2.7.3 Recall (Sensitivity)

Measures the extent to which the model can identify all true positive instances. Recall is calculated using equation (5).

$$\text{Recall} = \frac{\text{True Postive}}{\text{True Postive} + \text{False Negative}} \quad (5)$$

The recall metric is calculated by dividing the number of correct positive cases (True Positive) by the total number of positive cases (True Positive + False Negative).

2.7.4 F1-Score

It is a combination of precision and recall, and gives an overall picture of the model's performance. F1-score can be calculated by using equation (6).

$$\text{F1 Score} = 2 \text{ X } \frac{\text{Presisi X Recall}}{\text{Presisi+Recall}} \quad (6)$$

The F1 score is calculated as the harmonic mean of the precision and recall scores. A higher F1 score denotes a better quality classifier. The F1 score is an important metric, especially in dealing with imbalanced data or when you need to balance the trade-off between precision and recall.

2.8 Model Implementation

The selected model will be implemented into a prototype dashboard system. Flask framework was used to build the web-based dashboard. Flask is a micro-framework designed to build web applications rapidly and its web framework used by Python. Flask is often used for prototyping frameworks. Therefore, Flask provides complete flexibility for adding requirements [25].

The prototype dashboard system consists of input, process, and output, as shown in Figure. 5. The system input is new, unlabeled review data in the CSV format. The selected machine learning model then performs aspect-based sentiment classification processing in the process section. Meanwhile, the system output will display information on a dashboard page.

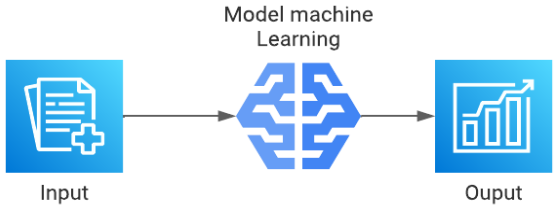


Figure 5. Prototype Dashboard System Framework

According to Wajong, a dashboard is a medium that can display information visually through tables and visual indicators [26]. In this study, the dashboard was designed to provide the following insights:

a. Which aspects received more positive or negative reviews?
b. Which keywords appear most often in each aspect?
c. How many reviews are there for each aspect?
d. What is the monthly sentiment trend?

Based on the system requirements, the researcher then created a prototype page in the form of a visual mock-up design of the input page (Figure. 6) and output page (Figure. 7) using wire-frames. A wire-frame is a schematic or blueprint of the structure of the software or website to be created [27].

Figure 6. Input Page

Figure. 6 shows the design of the input page as well as the homepage of the prototype system, this design depicts an HTML form with input elements to upload review data files and a button to initiate aspect-based sentiment analysis.

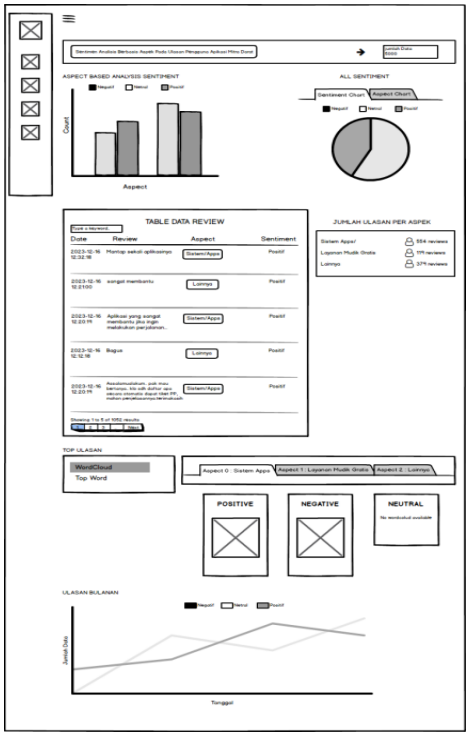


Figure 7. Dashboard Page

Figure. 7 shows the wireframe design of the output page showing the results of aspect-based sentiment analysis of *Mitra Darat* user reviews. The results are presented in the form of a dashboard.

3. RESULT AND DISCUSSION

3.1 Datasets

This study used 980 reviews, and the data distribution from the labeling process is shown in Table 3 and Table 4.

Table 3. Distribution of aspects

No	Aspect	Number
1	System/Apps	454
2	Service (Free *Mudik*)	191
3	Others	335
	Total	980

Table 3 presents the results of the aspect labeling process. Out of 980 review data, 454 reviews belong to the system aspect category, 191 reviews to the service aspect, and 335 reviews to other aspects.

Table 4. Distribution of datasets

No	Aspect	Positive	Neutral	Negative
1	System/Apps	69	59	326
2	Service (Free *Mudik*)	105	69	17
3	Others	314	7	14
	Total	488	135	357

Table 4 displays the distribution of data for each aspect along with the sentiment polarity. There are 488 reviews of positive sentiment, 357 reviews of negative sentiment, and 135 reviews of neutral sentiment.

3.2 Pre-processing

The pre-processing process cleaned the data through a number of stages such as case folding, cleansing and tokenization, normalization, negation handling, and stop-words removal. Examples of the results of the data pre-processing are shown in Table 5. After data pre-processing, 961 clean data points remained following the removal of empty data.

Table 5. Data preprocessing

Pre-processing	User Review (Before)	User Review (After)
Case folding	*Aplikasi ga jelas. Ga bisa di buka. Klo aplikasi pemerintah kenapa jelek yak ?*	*aplikasi ga jelas. ga bisa di buka. klo aplikasi pemerintah kenapa jelek yak ?*
Cleansing	*aplikasi ga jelas. ga bisa di buka. klo aplikasi pemerintah kenapa jelek yak ?*	*aplikasi ga jelas ga bisa di buka klo aplikasi pemerintah kenapa jelek yak*
Tokenising	*aplikasi ga jelas ga bisa di buka klo aplikasi pemerintah kenapa jelek yak*	*['aplikasi', 'ga', 'jelas', 'ga', 'bisa', 'di', 'buka', 'klo', 'aplikasi', 'pemerintah', 'kenapa', 'jelek', 'yak']*
Normalization	*['aplikasi', 'ga', 'jelas', 'ga', 'bisa', 'di', 'buka', 'klo', 'aplikasi', 'pemerintah', 'kenapa', 'jelek', 'yak']*	*['aplikasi', 'tidak', 'jelas', 'tidak', 'bisa', 'di', 'buka', 'kalau', 'aplikasi', 'pemerintah', 'kenapa', 'jelek', 'ya']*
Negation Handling	*['aplikasi', 'tidak', 'jelas', 'tidak', 'bisa', 'di', 'buka', 'kalau', 'aplikasi', 'pemerintah', 'kenapa', 'jelek', 'ya']*	*['aplikasi', 'tidak', 'tidak_jelas', 'tidak', 'tidak_bisa', 'di', 'buka', 'kalau', 'aplikasi', 'pemerintah', 'kenapa', 'jelek', 'ya']*
Stopword removal	*['aplikasi', 'tidak', 'tidak_jelas', 'tidak', 'tidak_bisa', 'di', 'buka', 'kalau', 'aplikasi', 'pemerintah', 'kenapa', 'jelek', 'ya']*	*['aplikasi', 'tidak_jelas', 'tidak_bisa', 'buka', 'aplikasi', 'pemerintah', 'jelek']*
Untokenisation and Remove Underscore	*['aplikasi', 'tidak_jelas', 'tidak_bisa', 'buka', 'aplikasi', 'pemerintah', 'jelek']*	*aplikasi tidak jelas tidak bisa buka aplikasi pemerintah jelek*

Table 5 shows the data changes at each stage of pre-processing, from casefolding to stop word removal and restored to its original form.

3.3 Word Embedding

A pre-trained model from IndoBert-pase-p was used for word embedding training. Figure. 8 shows an example of the word embedding results.

	text	embeddings	sentiment
0	tanggal maret sistem down pesan solusinya min	[1.5843604, 0.43504, 1.1427729, -0.22013655, 0...	Negatif
1	buka aplikasi eror	[0.3040867, 1.9236228, 1.9061613, 0.69554794, ...	Negatif
2	pelayanan bagus	[0.3851549, 0.5972539, 0.8466136, -0.43910268,...	Positif
3	membantu banget alhamdulillah terima kasih ada...	[0.4942519, 1.2218007, 0.6159986, 0.90845037, ...	Positif
4	oke mantap	[0.1766997, 1.3984344, 1.7110877, 1.1960614, -...	Positif
...	...	...	...

Figure 8. Word Embedding Result

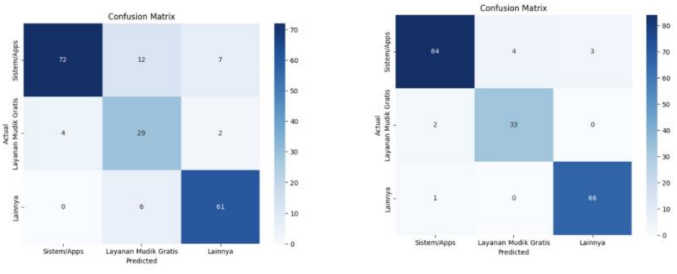
Figure. 8 displays how a sentence is transformed into numerical data (vector) through the word embedding process. This numeric data is what will be recognized by the machine learning algorithm in the classification process.

3.4 Result from ABSA Modelling

There were two classification processes in the machine learning modelling stage: an aspect classification model and a sentiment classification model. Each algorithm (Naive Bayes and Support Vector Machine) was trained and tested for both classifications. Models were evaluated using the confusion matrix.

3.4.1 Aspect Classification

The confusion matrix results from the aspect classification tests using Naive Bayes and Support Vector Machine with 193 test data (20%) can be seen in Figure. 9.



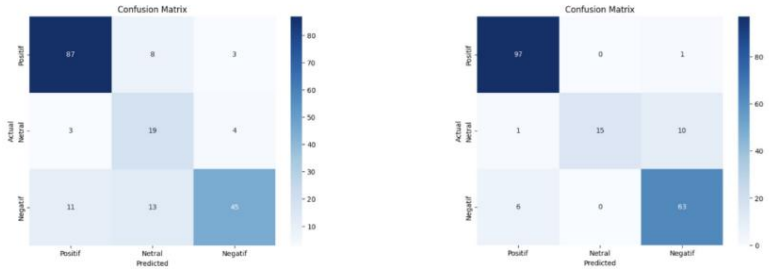
(a) (b)

Figure 9. Confusion Matrix Aspect Classification (a) Naive Bayes (b) SVM

Figure. 9 shows that for the aspect classification task using Naive Bayes, the correct prediction results for the system aspect is 72 data, the service aspect is 29 data, and 61 data for the other aspect. Meanwhile, with the Support Vector Machine, the correct prediction results for the system aspect is 84 data, the service aspect is 33 data, and the other aspect is 66 data. The number of wrong predictions from using Naive Bayes is 31 data and the number of wrong predictions from using the Support Vector Machine is 10 data.

3.4.2 Sentiment Classification

The confusion matrix results for sentiment classification testing using Naive Bayes and Support Vector Machine with 193 test data (20%) can be seen in Figure. 10.



(a) (b)

Figure 10. Confusion Matrix Sentiment Classification (a) Naive Bayes (b) SVM

Figure. 10 shows the results from the sentiment classification task using Naive Bayes. The correct prediction results for positive sentiment is 87 data, neutral sentiment is 19 data, and negative sentiment is 45 data. Meanwhile, when the Support Vector Machine algorithm was used, the correct prediction results for positive sentiment is 97 data, neutral sentiment is 15, and negative sentiment is 65 data. The number of wrong predictions using Naive Bayes is found in 45 data, and the number of wrong predictions using the Support Vector Machine is 18 data.

3.5 Model Selection

The comparison results of Naive Bayes and Support Vector Machine methods are shown in Table 6 for aspect classification and Table 7 for sentiment classification.

Table 6. Evaluation of the aspect classification model

Algorithm	Precision	Recall	F1-Score	Accuracy
Naive Bayes	86%	84%	84%	84%
SVM	**94%**	**94%**	**94%**	**94%**

Table 6 indicates that the SVM algorithm outperforms the Naïve Bayes algorithm in aspect classification, with an accuracy of 94% compared to 86%.

Table 7. Evaluation of the sentiment classification models

Algorithm	Precision	Recall	F1-Score	Accuracy
Naive Bayes	81%	78%	78%	78%
SVM	**91%**	**91%**	**90%**	**90%**

Table 7 indicates that the SVM algorithm also outperforms the Naïve Bayes algorithm for sentiment classification, with an accuracy of 91% compared to 81%.

Based on the model evaluation table above, the aspect classification and sentiment classification models based on the Support Vector Machine are stored and selected as models for aspect classification and sentiment classification in the prototype aspect-based sentiment analysis system for user reviews.

3.6 Model Implementation

The selected machine learning model is then implemented in a web-based application using the Flask framework. The application will analyze unlabeled review data and then display the analysis results through a dashboard. The prototype dashboard displaying the 2023 user review data analysis results for the *Mitra Darat* application is shown in Figure 11.

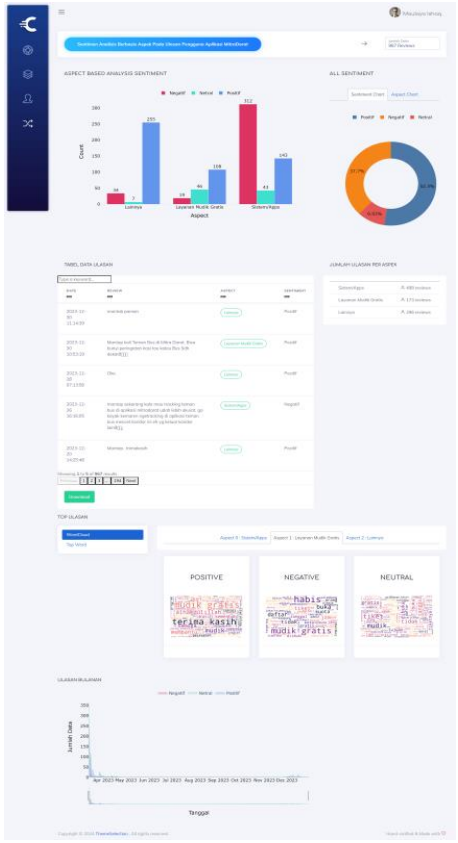


Figure 11. Output Dashboard Page

Figure. 11 displays the outcomes of the aspect-based sentiment analysis for the *Mitra Darat* user reviews. The results offer valuable insights, such as:

3.6.1 Which Aspects Received More Positive or Negative Reviews?

Figure. 12 shows the sentiment distribution information for each aspect. The system aspect received 312 positive, 43 neutral, and 143 negative reviews. The service aspect received 108 positive, 46 neutral, and 19 negative reviews. The other aspect received 255 positive, 7 neutral, and 34 negative reviews.

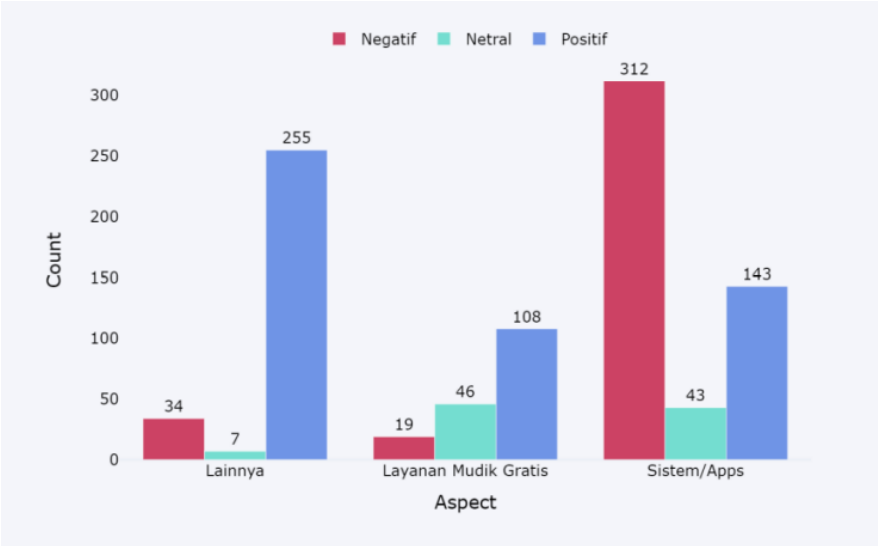


Figure 12. Sentiment Distribution for Each Aspect

3.6.2 Which Keywords Appear Most Often in Each Aspect?

Figure. 13, Fig. 14, Fig. 15 shows visualization with word clouds for each aspect to see which keywords often appear in user reviews of the *Mitra Darat* application.

Figure 13. World Cloud System Aspect

Figure. 13 shows a word cloud of the most frequent words in the system aspect. For the positive polarity the most frequent words are "aplikasi", "mantap", "membantu". For the negative polarity the words that often appear frequently are "tidak bisa", "login", "hypertext". For neutral sentiment the words that appear frequently are "min", "tiket", "habis".

Figure 14. World Cloud Service Aspect

Figure. 14 shows a word cloud of the most frequent words in the service aspect. For the positive polarity the words that appear frequently are "*mudik gratis*", "*terima kasih*" and "*alhamdulillah*". For the negative sentiment the words that appear frequently are "*mudik gratis*", "*habis*", "*daftar*". For neutral sentiment the words that appear include "*tiket*", "*habis*", "*min*".

Figure 15. World Cloud Other Aspect

Figure. 15 shows a word cloud with the most frequent words in the other aspect. For the positive polarity the words that appear frequently are "*bagus*", "*terima kasih*" and "*membantu*". For negative sentiment the words that appear frequently are "*bingung*", "*lambat*", "*susah*". For neutral sentiment the most frequent words are "*rute*", "*habis*", "*aduan*".

The obtained keywords were entered in the review data table to determine the review context, as shown in Fig. 16 with the example of the keyword "*tidak bisa*".

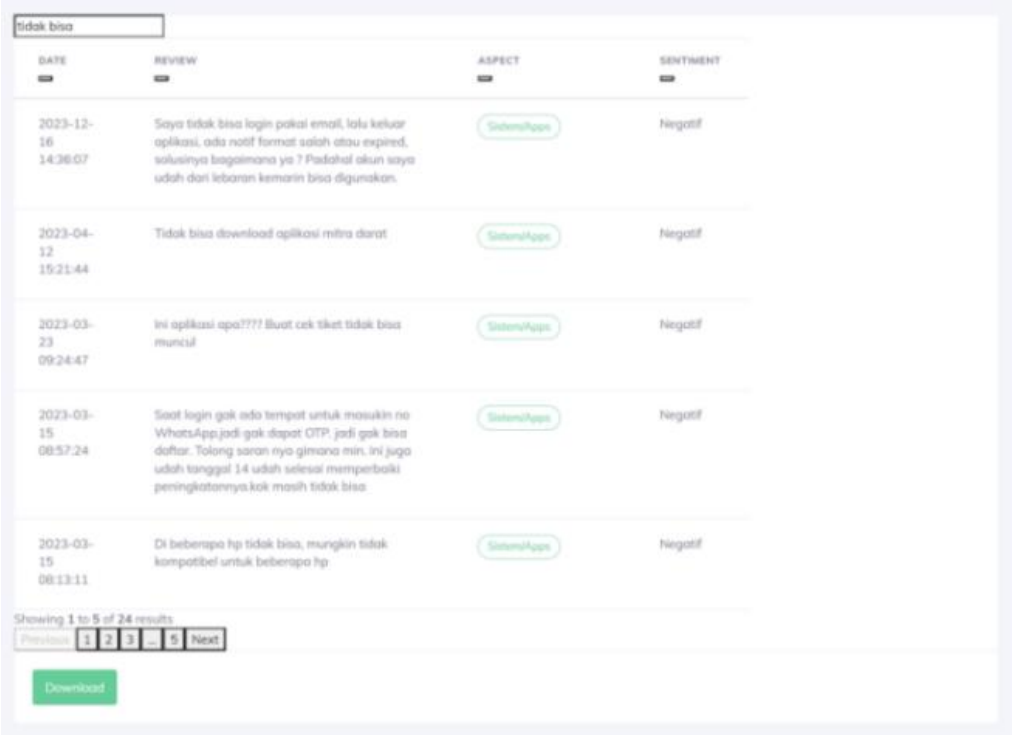


Figure 16. Result Tabel

Figure 16 shows the results of the data analysis, which includes aspect labels and sentiment presented in a table format. The table has a filter feature that allows users to search for reviews based on keywords associated with each aspect and sentiment polarity. For example, by entering the keyword "error" you can quickly find out the number of reviews containing the word "error" and access detailed information about those reviews.

3.6.3 How Many Reviews for Each Aspect?

Figure. 17 shows the number of reviews per aspect, which shows 498 reviews on the system aspect, 173 reviews on the free *mudik* service aspect, and 296 reviews on the other aspect with only short sentences. Meanwhile, for sentiment information, 52.3% or 506 reviews have positive sentiments, 37.7% or 365 reviews have negative sentiments, including 312 complaints about the system aspect (see Fig. 12), and 9.93% or 96 reviews have neutral sentiments.

Figure 17. Number of Reviews

3.6.4 What is the Monthly Sentiment Trend?

Figure. 18 shows that most reviews appear between April and May 2023 and in December 2023, as these are *mudik* periods. Sentiment trend information per month is difficult to determine due to the unbalanced distribution of data per month.

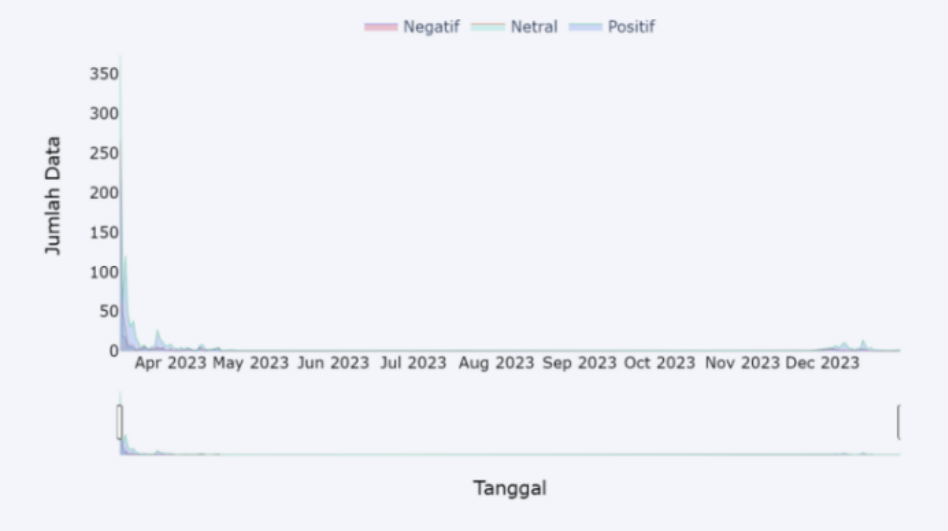


Figure 18. Monthly Sentiment Trend

4. CONCLUSION

Based on the comparison of the Naive Bayes algorithm and the Support Vector Machine (SVM), it was found that the SVM algorithm has a better performance by obtaining an accuracy value of 94% for the classification aspect and 90% for the sentiment classification, compared to the Naive Bayes algorithm which obtained an accuracy value of 84% for the classification aspect and 78% for the sentiment classification. After the model was implemented in the system dashboard prototype with the input of the user review of the *Mitra Darat* application in the period January - December 2023, it showed that the overall user sentiment was 52.3% positive with 506 reviews, 10% neutral with 96 reviews and, 37.7% negative with 365 reviews. The sentiment polarity by aspect shows that the system aspect is dominated by negative sentiment with 63%, with 8% neutral and 8 % positive. Conversely, the service aspect was 62% positive, 27% neutral, and 11% negative. This indicates that the system aspect requires improvement to address existing bugs, whereas the service aspect is quite well organized. Consequently, the application owner, the Directorate General of Land Transportation of the Ministry of Transportation, should prioritize enhancing the application's reliability and fixing system bugs. Future research on these datasets should consider incorporating methods for handling unbalanced data and applying a multi-label classification approach.

REFERENCES

[1] R. L. Wuri, “Biar Enggak Bingung Saat Mudik, Kemenhub Ajak Masyarakat Gunakan Aplikasi MitraDarat dan Ferizy,” wartaekonomi. Accessed: Dec. 08, 2023. [Online]. Available: https://wartaekonomi.co.id/read491399/biar-enggak-bingung-saat-mudik-kemenhub-ajak-masyarakat-gunakan-aplikasi-mitradarat-dan-ferizy

[2] M. Arnani, L. A. Azanella, and I. D. Wedhaswary, “Mudik, ‘Mulih Dhisik’, Kembali ke Udik...,” Kompas.com, Jun. 07, 2018. Accessed: Dec. 08, 2023. [Online]. Available: https://nasional.kompas.com/read/2018/06/07/09311731/mudik-mulih-dhisik-kembali-ke-udik

[3] S. Panichella, A. Di Sorbo, E. Guzman, C. A. Visaggio, G. Canfora, and H. C. Gall, “How can i improve my app? Classifying user reviews for software maintenance and evolution,” in 2015 IEEE International Conference on Software Maintenance and Evolution (ICSME), IEEE, 2015, pp. 281–290. doi: 10.1109/ICSM.2015.7332474.

[4] S. Venkatakrishnan, A. Kaushik, and J. K. Verma, “Sentiment Analysis on Google Play Store Data Using Deep Learning,” in Algorithms for Intelligent Systems, Springer Nature Singapore Pte Ltd, 2020, pp. 15–30. doi: 10.1007/978-981-15-3357-0_2.

[5] U. Pathak and Er. P. Rai, “Sentiment Analysis: Methods, Applications, and Future Directions,” Int J Res Appl Sci Eng Technol, vol. 11, no. 2, pp. 1453–1458, 2023, doi: 10.22214/ijraset.2023.49165.

[6] S. S. Shah, “Opinion Mining For Text Data: An Overview,” Int J Res Appl Sci Eng Technol, vol. 10, no. 6, pp. 4301–4311, 2022, doi: 10.22214/ijraset.2022.44902.

[7] M. Wankhade, A. C. S. Rao, and C. Kulkarni, “A survey on sentiment analysis methods, applications, and challenges,” Artif Intell Rev, vol. 55, no. 7, pp. 5731–5780, 2022, doi: 10.1007/s10462-022-10144-1.

[8] S. N. Hakim, A. J. Putra, and A. U. Khasanah, “Sentiment analysis on myindihome user reviews using support vector machine and naïve bayes classifier method,” International Journal of Industrial Optimization, vol. 2, no. 2, pp. 151–164, 2021, doi: 10.12928/ijio.v2i2.4437.

[9] E. Indrayuni, A. Nurhadi, and D. A. Kristiyanti, “Implementasi Algoritma Naive Bayes, Support Vector Machine, dan K-Nearest Neighbors untuk Analisa Sentimen Aplikasi Halodoc,” Faktor Exacta, vol. 14, no. 2, pp. 64–71, 2021, doi: 10.30998/faktorexacta.v14i2.9697.

[10] Z. Nurthohari, D. I. Sensuse, and S. Lusa, “Sentiment Analysis of Jakarta Bus Rapid Transportation Services using Support Vector Machine,” in 2022 International Conference on Data Science and Its Applications (ICoDSA), IEEE, 2022, pp. 171–176. doi: 10.1109/ICoDSA55874.2022.9862903.

[11] C. A. Bahri and L. H. Suadaa, "Aspect-Based Sentiment Analysis in Bromo Tengger Semeru National Park Indonesia Based on Google Maps User Reviews," IJCCS (Indonesian Journal of Computing and Cybernetics Systems), vol. 17, no. 1, pp. 79–90, 2023, doi: 10.22146/ijccs.77354.
[12] M. S. Mubarok, Adiwijaya, and M. D. Aldhi, "Aspect-based sentiment analysis to review products using Naïve Bayes," in AIP conference proceedings, AIP Publishing, 2017, p. 020060. doi: 10.1063/1.4994463.
[13] R. K. Singh and K. Ramalingam, "Amazon Product Review Sentiment Analysis with Machine Learning," International Journal of Trend in Scientific Research and Development (ijtsrd), vol. 5, no. 4, pp. 720–723, 2021.
[14] A. M. Rahat, A. Kahir, and A. K. M. Masum, "Comparison of Naive Bayes and SVM Algorithm based on Sentiment Analysis Using Review Dataset," in 2019 8th International Conference System Modeling and Advancement in Research Trends (SMART), IEEE, 2019, pp. 266–270. doi: 10.1109/SMART46866.2019.9117512.
[15] D. I. Af'idah, P. D. Anggraeni, M. Rizki, A. B. Setiawan, and S. F. Handayani, "Aspect-Based Sentiment Analysis for Indonesian Tourist Attraction Reviews Using Bidirectional Long Short-Term Memory," JUITA: Jurnal Informatika, vol. 11, no. 1, pp. 27–36, 2023.
[16] G. Dwyer, S. Aggarwal, and J. Stouffer, Flask: building python web services. Packt Publishing, 2017.
[17] V. Gupta and G. S. Lehal, "A survey of text mining techniques and applications," Journal of Emerging Technologies in Web Intelligence, vol. 1, no. 1, pp. 60–76, 2009.
[18] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification," Inf Process Manag, vol. 50, no. 1, pp. 104–112, 2014, doi: 10.1016/j.ipm.2013.08.006.
[19] E. Yulianti, N. Pangestu, and M. A. Jiwanggi, "Enhanced TextRank using weighted word embedding for text summarization," International Journal of Electrical and Computer Engineering (IJECE), vol. 13, no. 5, pp. 5472–5482, 2023, doi: 10.11591/ijece.v13i5.pp5472-5482.
[20] B. Wilie et al., "IndoNLU: Benchmark and resources for evaluating Indonesian natural language understanding," in Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, 2020.
[21] J. Hartmann, M. Heitmann, C. Siebert, and C. Schamp, "More than a Feeling: Accuracy and Application of Sentiment Analysis," International Journal of Research in Marketing, vol. 40, no. 1, pp. 75–87, 2023, doi: 10.1016/j.ijresmar.2022.05.005.
[22] Scikit-learn, "Naive Bayes," Scikit-learn. [Online]. Available: https://scikit-learn.org/stable/modules/naive_bayes.html
[23] V. N. Vapnik, "An overview of statistical learning theory," IEEE Trans Neural Netw, vol. 10, no. 5, pp. 988–999, 1999, doi: 10.1109/72.788640.
[24] D. Gil and M. Johnsson, "Support vector machines in medical classification tasks," in Support Vector Machines: Data Analysis, Machine Learning and Applications, Nova Science Publishers, Inc., 2011, pp. 81–102.
[25] D. Ghimire, "Comparative study on Python web frameworks: Flask and Django," Metropolia University of Applied Sciences, 2020.
[26] A. M. R. Wajong, "Applying performance dashboard in hospitals," International Journal Of Software Engineering and Its Applications, vol. 9, no. 1, pp. 213–220, 2015, doi: 10.14257/ijseia.2015.9.1.19.
[27] P. Guilizzoni, "What are wireframes and why are they used?," Balsamiq Studios, LLC. Accessed: Dec. 17, 2023. [Online]. Available: https://balsamiq.com/learn/articles/what-are-wireframes/