Implementation of Naive Bayes Classifier with SMOTE for Sentiment Analysis of Blibli App Reviews on the Google Play Store

Rafli Hudanul Sidiq\*1, Azzam Dicky Umar Widadi2, and Afiftha Ravi Aufa Yubiharto3

1,2,3Informatics, Universitas Jenderal Soedirman, Indonesia

1H1D022069\_Rafli Hudanul Sidiq, 2H1D022090\_Azzam Dicky Umar Widadi, 3H1D022095\_Afiftha Ravi Aufa Yubiharto

Email: 1rafli.sidiq@mhs.unsoed.ac.id, 2azzam.widadi@mhs.unsoed.ac.id, 3afiftha.yubiharto@mhs.unsoed.ac.id

Received: Jun 9, 2024; Revised: Jun 9, 2024; Accepted: Jun 9, 2024; Published: Jun 9, 2024

Abstrak

|  |
| --- |
| In the digital age, online shopping has become prevalent, with platforms like the Google Play Store enabling users to download and review mobile applications. This study aims to analyze the sentiment of user reviews for the Blibli application on the Google Play Store using the Naive Bayes Classifier, a simple yet effective algorithm for text classification tasks. A total of 2500 recent reviews were scraped using Google Colaboratory and the Python programming language. Data preprocessing steps included cleaning, stopword removal, tokenization, and stemming, followed by addressing class imbalance using the Synthetic Minority Over-sampling Technique (SMOTE). The dataset was divided into training and testing sets in an 80:20 ratio. The Naive Bayes algorithm with SMOTE was employed for sentiment classification, yielding an accuracy of 90%, precision of 90%, recall of 92%, and an F1-score of 91%. These results demonstrate the model's reliability in distinguishing between positive and negative sentiments, with a slight bias towards positive sentiments. Additionally, word cloud visualizations were generated to highlight frequently occurring words in both positive and negative reviews. The findings provide valuable insights for Blibli application developers and stakeholders, aiding in the assessment of user satisfaction and identification of areas for improvement. This research underscores the efficacy of the Naive Bayes Classifier in sentiment analysis and the utility of Google Colaboratory for data processing tasks.  *Keywords: Analysis Sentiment, Blibli, Google Play Store, Naive Bayes, Python, SMOTE* |
| This work is an open access article and licensed under a Creative Commons Attribution-Non Commercial 4.0 International License |

1. **INTRODUCTION**

Shopping is one of the things that people do quite often. Goods purchased start from basic needs to goods for self-satisfaction. Currently, shopping is easy to do, namely by shopping online. This online shopping is very helpful for people in getting goods that are far from the location where they live. in online shopping there are also various attractive promotions so that people prefer to shop online compared to coming to the store directly. now online shopping platforms are diverse. these platforms can be easily downloaded such as the google play store.

In the ever-evolving digital age, mobile-based applications have become an integral part of everyday life. One of the largest platforms for mobile app distribution is the Google Play Store, which allows users to not only download apps, but also leave reviews and ratings. One of the features found in the Play Store is the rating and review feature where users of products from the Play Store can give their opinions on the products they have used [1]. These user reviews are an invaluable source of information for app developers as they provide insights into user satisfaction, app performance, and suggestions for improvement. Business companies generally use it to detect sentiment in social data, gage brand reputation, and understand the customer and their needs [2]. Lots of users use various online resources to express their views and opinions [3]. However, with such many reviews, manual analysis becomes an almost impossible task. Hence, an automated approach is required to analyze the sentiment of these reviews. Analyzing the sentiment tendency of consumer evaluation can not only provide a reference for other consumers but also help businesses on e-commerce platforms to improve service quality and consumer satisfaction [4].

Sentiment analysis focuses on identifying and extracting opinions or sentiments from text. This technique is used to determine whether text is positive and negative. One of the popular algorithms used for sentiment analysis is the Naive Bayes Classifier. This algorithm is known for its simplicity and good performance in various text classification tasks.

The Naïve Bayes method is a classification method in machine learning that excels in using training data samples toestimate parameters involved in the classification process rapidly, resulting in high accuracy [5]. based on previous research written by [5] entitled “The Shopee Application User Reviews Sentiment Analysis Employing Naïve Bayes Algorithm” concluded that the sentiment analysis of user reviews using naïve bayes with training data and 8:2 test data gets an accuracy of 86.00% which shows superior positive sentiment analysis results. In the previous analysis conducted by [6] with the title “Sentiment analysis of game product on shopee using the TF-IDF method and naive bayes classifier” get the results of sentiment analysis which can be concluded that Sentiment analysis using TF-IDF method and Naive Bayes Classifier based on reviews from buyers (regardless of rating feature). The data collected were 1000 game product reviews on the online shopping site Shopee, divided into 700 training data and 300 test data. Based on the research results, the accuracy rate is 80.2223% and the f1 value is 0.691372. Then from another journal analysis with the title “App Review Sentiment Analysis Shopee Application In Google Play Store Using Naive Bayes Algorithm” written by [7] with 200 review data taken consisting of 100 positive reviews and 100 negative reviews applying data mining to its analysis with naive bayes with partitioning techniques resulting in 96.667% accuracy, 100% precision, 93.33% recall, and AUC 1.00 which can be said to be included in a very good classification. from other research in other journals with the title "Sentiment Analysis of Online Transportation Service using the Naïve Bayes Methods" written by [8] obtained the final result using the naive Bayes method for data mining or text mining classification, namely 81.00% accuracy.

In this study, a sentiment analysis of user reviews on the Blibli application on the google play store using naive bayes will be carried out by taking user review data using google colab. Collaboratory, or Colab for short, is a Google Research tool that allows developers to write and run Python code in their browser. Google Colab is an amazing tool for hands-on learning activity. It is a little Jupyter notebook that requires no installation and includes a fantastic free edition that gives you free access to Google computer resources like GPUs and TPUs [9]. around 2500 data which will be divided for testing data and training data. The benefits of this research are expected to be able to find out the response of application users conveyed through opinions that are both positive and negative.

1. METHOD
   1. Problem Formulation

This study aims to analyze the sentiment of user reviews for the Blibli application on the Google Play Store. The primary objective is to classify reviews as positive or negative using the Naive Bayes Classifier. This method leverages the simplicity and efficiency of the Naive Bayes algorithm in handling text classification tasks. The research stages are outlined to ensure a systematic approach, including data collection, preprocessing, model training, and evaluation.

Sentiment analysis, a crucial aspect of natural language processing (NLP), involves determining the emotional tone behind a series of words. In this case, it helps in understanding customer feedback by categorizing reviews into positive or negative sentiments. This classification aids the Blibli application developers and stakeholders in assessing user satisfaction and identifying areas for improvement. Sentiment analysis is one of the key analyses that is currently used with the aim of classifying sentiments and opinions generated by human beings and in text [10].

* 1. Naïve Bayes Classifier

In the research of text mining, document classification is a growing field. Even though we have many existing classifying approaches, Naïve Bayes Classifier is simple and effective at classification. At this stage the data are analyzed, then models are applied according to the type of data. The model proposed in this study is Naive Bayes.

The Naive Bayes Classifier is based on Bayes' theorem, which provides a way of calculating the posterior probability, from the prior probability, and the likelihood. The theorem is formulated as follows:

(1)

= This is the probability of class given the data .

= This is the prior probability of class .

= This is the likelihood of the data given that it is from class .

= This is the prior probability of the data .

The "naive" aspect of the classifier comes from the assumption that the features (in this case, words in a review) are conditionally independent given the class. This assumption simplifies the computation significantly, making the algorithm very efficient, especially for large datasets.

* 1. Research Stages

The research involves several stages, each critical to the overall success of the sentiment analysis.

* + 1. Data Collection

Web scraping was performed using Google Colaboratory and the Python programming language. The Google-play-scraper library was used to scrape 2500 recent reviews of the Blibli application from the Google Play Store.

* + 1. Data Preprocessing

The following stage is preprocessing. Preprocessing is stage in which we clean the data before extracting its features. Preprocessing can help to avoid data interference, imperfect data, and inconsistent data [11]. Pre-processing is a process of changing the form of data that has not been structured into structured data as needed, for further mining processes [12].

1. Cleaning: Removal of noise and irrelevant data, particularly neutral value data.
2. Stopword Removal: Elimination of common words that do not contribute to the sentiment, based on a predefined stopword list.
3. Tokenizing: Breaking down text into individual tokens for easier processing.
4. Stemming: Reducing words to their root form using the “Sastrawi” library.
5. SMOTE (Synthetic Minority Over-sampling Technique): Used to address class imbalance by increasing the number of samples in the minority class.
   * 1. Data Splitting

The dataset was divided into training and testing sets using the hold-out method, with 80% for training and 20% for testing.

* + 1. Classification

1. The Naive Bayes algorithm, based on Bayes' theorem, was employed for classifying the sentiment of the reviews.
2. Model performance was evaluated using a confusion matrix, calculating accuracy, precision, recall, and F1-score.
   * 1. Visualization

In the context of this study, word clouds were generated to visualize the most frequently used words in positive and negative reviews. This visualization technique provides a clear and intuitive way to understand the prominent terms in user feedback, thereby aiding in the analysis and interpretation of the sentiment conveyed in the reviews. The size of the letters is determined by the intensity with which the word is used. The more often it is used, the larger the letter size of the word [13].

1. RESULT
   1. Web Scrapping

Web Scrapping which was carried out using Google Colaboratory tools and the Python programming language on the Blibli link on the Google Play Store obtained 2500 review data from the latest. This process is carried out by installing the Google-play-scraper library which is used to scrape review data on the Google Play Store by simply entering the ID of the application whose review you want to retrieve which is found in the application link [14]. The following is a dataset that has been taken and 5 examples of data are displayed which can be seen in figure 1.

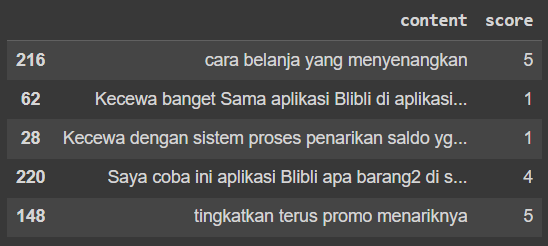


Fig 1**.** Scrapping Result

* 1. Preprocessing

Text preprocessing entails the removal of punctuation and symbols, the removal of stop words and slang, and stemming [15]. Data preprocessing, or data pre-processing, is a series of steps or stages carried out on raw data before the data is used for further analysis or model development. The main goal of data preprocessing is to improve data quality, ensure the accuracy of analysis results, and address problems or deficiencies that may arise in the raw data [16].

* + 1. Cleaning

Data cleaning is a process carried out to remove noise from data that is inconsistent or could be called irrelevant. For cleaning is to remove data that has a neutral value. The following are the results of data cleaning which can be seen in figure 2.

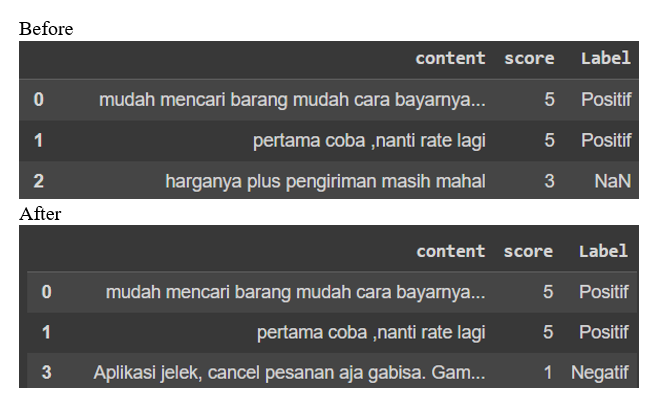


Fig 2. Data Cleaning

* + 1. Stopword Removal

Stopword Removal is part of the text preprocessing stage which aims to remove irrelevant words in a sentence based on the stopword list. The list of stop words that are commonly used is in the form of a digital library whose list is already available in advance [17]. Stop word removal can be thought of as an entity selection routine, in which entities that do not contribute to correct ranking decisions are considered spurious words and are removed from the entity space accordingly [18]. The following are the results of stopword removal which can be seen in the figure 3.

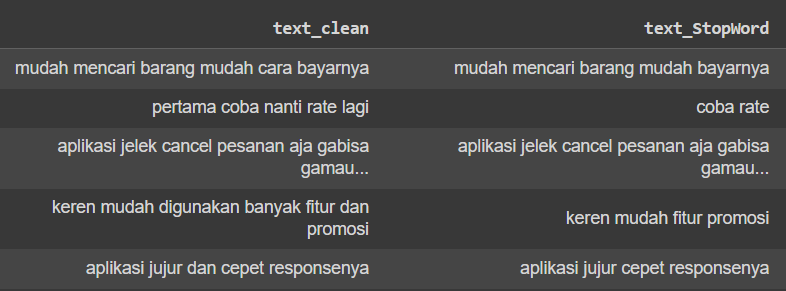


Fig 3. Stopword Removal

* + 1. Tokenizing

Tokenization is a procedure for recovering the elements of interest in a sequence of data. This term is commonly used to describe an initial step in the processing of programming languages, and also for the preparation of input data in the case of artificial neural networks [19]. Sentences are categorized into words or tokens by ignoring whitespaces and other symbols [20]. Tokenization refers to splitting the text into meaningful smaller units known as tokens [21]. The following are the results of tokenizing which can be seen in the figure 4.

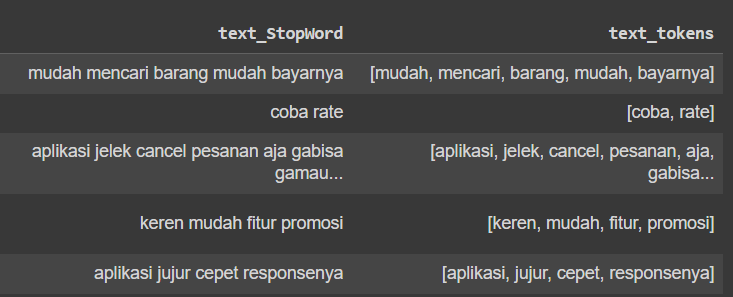


Fig 4. Tokenizing

* + 1. Stemming

Stemming is the next process after filtering to get root words from each token. The stemming method used in developing corpus was “Sastrawi” [22]. The following are the results of stemming which can be seen in the figure 5.

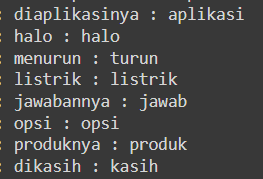


Fig 5. Stemming

* 1. Smote

SMOTE is used to increase the number of samples in the minority class [23]. The SMOTE (synthetic Minority Over-sampling Technique) oversampling method used to deal with class imbalance problems [24]. The following are the results of smote which can be seen in the figure 6 and 7.

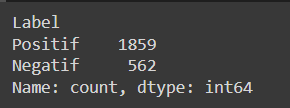


Fig 6. Before Smote

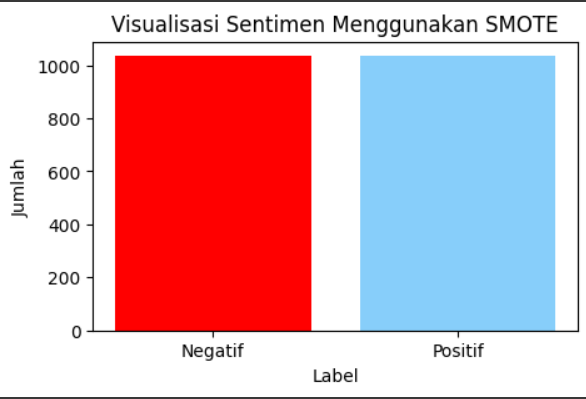


Fig 7. Smote

* 1. Splitting Data

The process of dividing training data using the hold-out method divides training data by 80% and testing data by 20%. The data-sharing process uses the Python programming language which can be seen in the figure 8.

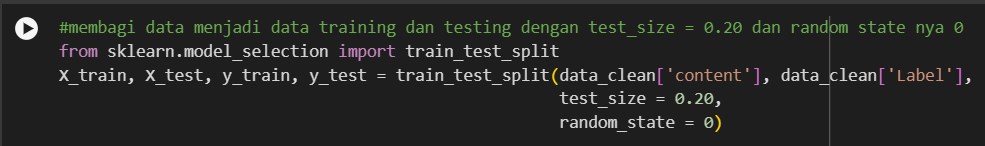


Fig 8. Splitting Data

* 1. Classification

Naïve Bayes technique is utilized for both categorization and training. Testing using the confusion matrix was carried out to test the model implemented on training data and testing data. Based on the test results using the confusion matrix, the results are shown in the table 1 and 2.

Table 1. Confusion Matrix Result

|  |  |  |
| --- | --- | --- |
| Prediction Data | Actual Data | |
| Positive | Negative |
| Positive | 96 | 21 |
| Negative | 23 | 349 |
|  |  |  |

Table 2. Confusion Matrix Result with SMOTE

|  |  |  |
| --- | --- | --- |
| Prediction Data | Actual Data | |
| Positive | Negative |
| Positive | 249 | 25 |
| Negative | 19 | 165 |

The results of the naive Bayes algorithm obtain accuracy, precision, recall and f1-score values ​​which can be seen in the figure 9 and 10.

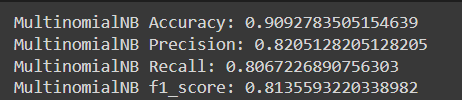


Fig 9. Accuracy, Precision, Recall, and f1-score Results

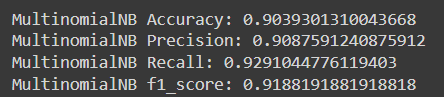


Fig 10. Accuracy, Precision, Recall, and f1-score Results with SMOTE

The following are the sentiment results from the Blibli application review on Google Playstore, the results of which can be seen in the figure 11 and 12.

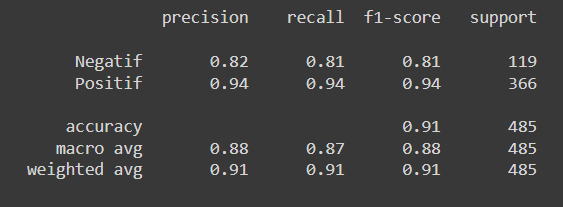


Fig 11. Sentiment Results

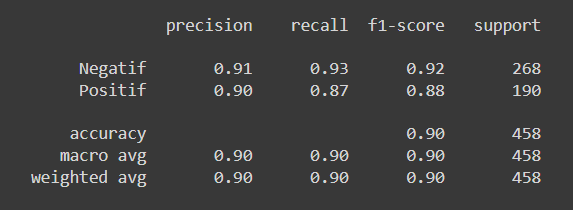


Fig 12. Sentiment Results with SMOTE

* 1. Visualization

In this research, the outputs are word clouds from a visual stage [25]. The images in the form of words that are frequently used in each expression will be shown. The following is a visualization of words that are often used in negative reviews as well as positive reviews that can be seen in figures 13 and figures 14.



Fig 13. Visualization of Negative Words



Fig 14. Visualization of Positive Words

* 1. Testing

The testing used uses a python language program which if input a review, the program will determine whether the review is positive or negative. The following are the results of testing which can be seen in the table 3.

Table 3. Testing Result

|  |  |
| --- | --- |
| Review | Sentiment |
| Aplikasi ini bagus dan lancar digunakan | Positive |
| Aplikasi yang mudah dan banyak promo | Positive |
| Belanja ribet dan saya kecewa tidak bisa batal pesan | Negative |

1. discussions

The results of the study provide insights into the sentiment analysis of Blibli application reviews from the Google Play Store. Through a comprehensive web scraping and data preprocessing workflow, extract, clean, and analyze 2500 reviews to determine user sentiment.

Sentiment analysis using the Naive Bayes algorithm yielded promising results, with an accuracy rate indicating that the model correctly classified a significant portion of the reviews. Specifically, the confusion matrix showed 96 true positive predictions and 345 true negative predictions, alongside 23 false positives and 21 false negatives then with SMOTE showed 249 true positive predictions and 165 true negative predictions, alongside 19 false positives and 25 false negatives. These results demonstrate the model's capability to distinguish between positive and negative sentiments effectively.

The accuracy, precision, recall, and F1-score metrics further validate the model's performance. With an accuracy of 0.90, a precision of 0.82, a recall of 0.80, and an F1-score of 0.81 then use SMOTE with an accuracy of 0.90, a precision of 0.90, a recall of 0.92, and an F1-score of 0.91, the model demonstrates strong reliability in predicting both positive and negative sentiments. These metrics indicate that the model is particularly effective at identifying positive sentiments, as reflected in the high recall value. This slight bias towards positive sentiment predictions could be attributed to the inherent nature of app reviews, where satisfied users are more likely to leave reviews than dissatisfied users.

1. CONCLUSION

This study conducted sentiment analysis on reviews of the Blibli application from the Google Play Store using web scraping via Google Colaboratory and Python. A total of 2500 recent reviews were collected and processed through various data preprocessing stages, including cleaning, stopword removal, tokenization, and stemming. The data were then balanced using SMOTE to ensure they were more balanced and usable. The Naive Bayes algorithm was used for classification, resulting in an accuracy of 90%, precision of 90%, recall of 92%, and an F1-score of 91%, demonstrating the model's effectiveness in distinguishing between positive and negative sentiments. Word cloud visualization helped identify frequently occurring words in positive and negative reviews.

REFERENCES

[1] A. T. Rizkya, R. Rianto, and A. I. Gufroni, “Implementation of the Naive Bayes Classifier for Sentiment Analysis of Shopee E-Commerce Application Review Data on the Google Play Store,” *Int. J. Appl. Inf. Syst. Informatics*, vol. 1, no. 1, pp. 31–37, 2023.

[2] M. Demircan, A. Seller, F. Abut, and M. F. Akay, “Developing Turkish sentiment analysis models using machine learning and e-commerce data,” *Int. J. Cogn. Comput. Eng.*, vol. 2, no. October, pp. 202–207, 2021, doi: 10.1016/j.ijcce.2021.11.003.

[3] M. Wankhade, A. C. S. Rao, and C. Kulkarni, *A survey on sentiment analysis methods, applications, and challenges*, vol. 55, no. 7. Springer Netherlands, 2022. doi: 10.1007/s10462-022-10144-1.

[4] L. Yang, Y. Li, J. Wang, and R. S. Sherratt, “Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning,” *IEEE Access*, vol. 8, pp. 23522–23530, 2020, doi: 10.1109/ACCESS.2020.2969854.

[5] N. A. Pasaribu and Sriani, “The Shopee Application User Reviews Sentiment Analysis Employing Naïve Bayes Algorithm,” *Int. J. Softw. Eng. Comput. Sci.*, vol. 3, no. 3, pp. 194–204, 2023, doi: 10.35870/ijsecs.v3i3.1699.

[6] R. Kosasih and A. Alberto, “Sentiment analysis of game product on shopee using the TF-IDF method and naive bayes classifier,” *Ilk. J. Ilm.*, vol. 13, no. 2, pp. 101–109, 2021, doi: 10.33096/ilkom.v13i2.721.101-109.

[7] D. Pratmanto, R. Rousyati, F. F. Wati, A. E. Widodo, S. Suleman, and R. Wijianto, “App Review Sentiment Analysis Shopee Application in Google Play Store Using Naive Bayes Algorithm,” *J. Phys. Conf. Ser.*, vol. 1641, no. 1, 2020, doi: 10.1088/1742-6596/1641/1/012043.

[8] M. Tika Adilah, H. Supendar, R. Ningsih, S. Muryani, and K. Solecha, “Sentiment Analysis of Online Transportation Service using the Naïve Bayes Methods,” *J. Phys. Conf. Ser.*, vol. 1641, no. 1, 2020, doi: 10.1088/1742-6596/1641/1/012093.

[9] M. Srivastava, V. Yadav, and S. Singh, “Implementation of Web Application for Disease Prediction Using AI,” *BOHR Int. J. Data Min. Big Data*, vol. 1, no. 1, pp. 5–9, 2020, doi: 10.54646/bijdmbd.002.

[10] P. P. A., “Performance Evaluation and Comparison using Deep Learning Techniques in Sentiment Analysis,” *J. Soft Comput. Paradig.*, vol. 3, no. 2, pp. 123–134, 2021, doi: 10.36548/jscp.2021.2.006.

[11] A. Rozaq, Y. Yunitasari, K. Sussolaikah, and E. R. N. Sari, “Sentiment Analysis of Kampus Mengajar 2 Toward the Implementation of Merdeka Belajar Kampus Merdeka Using Naïve Bayes and Euclidean Distence Methods,” *Int. J. Adv. Data Inf. Syst.*, vol. 3, no. 1, pp. 30–37, 2022, doi: 10.25008/ijadis.v3i1.1233.

[12] D. Nurcahyono, W. P. Putra, A. Najib, and T. R. Tulili, “Analysis sentiment in social media against election using the method naive Bayes,” *J. Phys. Conf. Ser.*, vol. 1511, no. 1, 2020, doi: 10.1088/1742-6596/1511/1/012003.

[13] A. Erfina and M. R. N. R. Alamsyah, “Implementation of Naive Bayes classification algorithm for Twitter user sentiment analysis on ChatGPT using Python programming language,” *Data Metadata*, vol. 2, pp. 2–11, 2023, doi: 10.56294/dm202345.

[14] N. Agustina, D. H. Citra, W. Purnama, C. Nisa, and A. R. Kurnia, “Implementasi Algoritma Naive Bayes untuk Analisis Sentimen Ulasan Shopee pada Google Play Store,” *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 2, no. 1, pp. 47–54, 2022, doi: 10.57152/malcom.v2i1.195.

[15] T. Hariguna and V. Rachmawati, “Community Opinion Sentiment Analysis on Social Media Using Naive Bayes Algorithm Methods,” *IJIIS Int. J. Informatics Inf. Syst.*, vol. 2, no. 1, pp. 33–38, 2019, doi: 10.47738/ijiis.v2i1.11.

[16] I. G. I. Sudipa, I. G. M. Darmawiguna, I. M. Dendi, and M. Sanjaya, *Buku ajar data mining*, no. January. 2024.

[17] A. T. Jaka Harjanta and B. A. Herlambang, “Extraction Sentiment Analysis Using naive Bayes Algorithm and Reducing Noise Word applied in Indonesian Language,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 835, no. 1, 2020, doi: 10.1088/1757-899X/835/1/012051.

[18] P. Thwe, Y. Y. Aung, and C. C. Lwin, “Naïve Bayes Classifier for Sentiment Analysis,” *Int. J. All Res. Writings*, vol. 3, no. 7, pp. 32–35, 2021, [Online]. Available: http://www.ijciras.com/Users/ManuScript/ManuScriptDetails/a9903b84-dfa2-406b-a9c4-7e2a060831c4

[19] R. Friedman, “Tokenization in the Theory of Knowledge,” pp. 380–386, 2023.

[20] R. V. B. Vangara\*, K. Thirupathur, and S. P. Vangara, “Opinion Mining Classification u sing Naive Bayes Algorithm,” *Int. J. Innov. Technol. Explor. Eng.*, vol. 9, no. 5, pp. 495–498, 2020, doi: 10.35940/ijitee.e2402.039520.

[21] M. N. Uddin, M. F. Bin Hafi, S. Hossain, and S. M. M. Islam, “Drug Sentiment Analysis using Machine Learning Classifiers,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 1, pp. 92–100, 2022, doi: 10.14569/IJACSA.2022.0130112.

[22] A. B. Mutiara, E. P. Wibowo, and P. I. Santosa, “Improving the accuracy of text classification using stemming method , a case of non ‑ formal Indonesian conversation,” *J. Big Data*, pp. 1–16, 2021, doi: 10.1186/s40537-021-00413-1.

[23] Z. Xu, D. Shen, T. Nie, and Y. Kou, “A hybrid sampling algorithm combining M-SMOTE and ENN based on Random forest for medical imbalanced data,” *J. Biomed. Inform.*, vol. 107, no. June, p. 103465, 2020, doi: 10.1016/j.jbi.2020.103465.

[24] Hermanto, A. Y. Kuntoro, T. Asra, E. B. Pratama, L. Effendi, and R. Ocanitra, “Gojek and Grab User Sentiment Analysis on Google Play Using Naive Bayes Algorithm and Support Vector Machine Based Smote Technique,” *J. Phys. Conf. Ser.*, vol. 1641, no. 1, 2020, doi: 10.1088/1742-6596/1641/1/012102.

[25] Y. Kalmukov, “Using Word Clouds for Fast Identification of Papers’ Subject Domain and Reviewers’ Competences 15,” *Proc. Univ. Ruse*, vol. 60, pp. 114–119, 2021.

##### **SUBMISSION CHECK-LIST**

All items must be YES before submitting this task in order to make sure a manuscript well-prepared within a high standard. Please put X in column Yes after checking each item

|  |  |  |  |
| --- | --- | --- | --- |
| **General Format** | |  |  |
| **No** | **Item** | **Yes** | **No** |
| 1 | I have prepared the manuscript using “DataMining-Template.docx” | YES |  |
| 2 | The manuscript is written in single or double column using Palatino Linotype10 pt font |  |  |
| 3 | Title is matching with the focus of the manuscript |  |  |
| 4 | I have complete authors information: |  |  |
|  | a. Affiliation of all authors |  |  |
|  | b. Indicate one of the authors is the corresponding author |  |  |
|  | c. Email of corresponding author |  |  |
| 5 | The length of the manuscript is not less than 2500 words (excluding abstract, author information and reference list) |  |  |
| 6 | The manuscript contains the core contents: introduction with literature review, method, results, discussion, conclusion, references |  |  |
| 7 | Contains maximum 2 level of sections |  |  |
|  |  |  |  |
| **Abstract** | |  |  |
| **No** | **Item** | **Yes** | **No** |
| 1 | The abstract is written between 150 – 250 words |  |  |
| 2 | contains the purpose |  |  |
| 3 | contains the method / research approach |  |  |
| 4 | contains the important findings / conclusion |  |  |
| 5 | contains novelty/originality |  |  |
| 6 | The abstract is completed with keywords |  |  |
|  |  |  |  |
| **Figures (if applicable)** | |  |  |
| **No** | **Item** | **Yes** | **No** |
| 1 | All figures are clear and readable (image, text and legend) with high resolution |  |  |
| 2 | All figures are in formal style, without redundant title |  |  |
| 3 | All information in the figure are in English and all decimal written in international standard, using point (.) not comma (,) |  |  |
| 4 | All figures have captions (at the bottom) with consecutive numbers |  |  |
| 5 | All figures are cited in the text using consistent citation style |  |  |
| 6 | Never citing figure using below and above, but using the number of figure |  |  |
| **Tables (if applicable)** | |  |  |
| **No** | **Item** | **Yes** | **No** |
| 1 | All texts in tables are clear and readable |  |  |
| 2 | All tables are in formal style |  |  |
| 3 | All information in the table are in English and all decimal written in international standard, using point (.) not comma (,) |  |  |
| 4 | All tables are captioned on top with consecutive numbers |  |  |
| 5 | All tables are cited in the text using consistent citation |  |  |
| 6 | Never citing table using below and above, but using the number of table |  |  |
| **Equations (if applicable)** | |  |  |
| **No** | **Item** | **Yes** | **No** |
| 1 | All equations are written using editor tool (editable), not a cropped image |  |  |
| 2 | All equations are captioned on top with consecutive numbers |  |  |
| 3 | All equations are cited in the text using consistent citation |  |  |
| 4 | Never citing equation using below and above, but using the number of equation |  |  |
| **References** | |  |  |
| **No** | **Item** | **Yes** | **No** |
| 1 | Reference list and citation consistently follows the IEEE style |  |  |
| 2 | Journal names in the reference list are not in abbreviated format, but it should be in full name format |  |  |
| 3 | All references are cited in the text |  |  |
| 4 | Citation in the text follows general consistent citation rules |  |  |
| 5 | Paper cites at least than 25 references |  |  |
| 6 | Book sources are not more than 20% of the reference list |  |  |
| 7 | Self-citation in the list is not more than 2 |  |  |
| 8 | Write all authors in the reference list unless authors more than 7  (Write the first 6, then et al.) |  |  |