Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th February 2025. Vol. 4. No. 2; e-ISSN: 2808-4519

Enhancing Model Accuracy in Sentiment Analysis of the by.U Application Using Naïve Bayes and SMOTE Techniques

Athhar Hafizha Luthfi[1]*, Ahmad Faqih[2], Gifthera Dwilestari[3]

[1,2]Teknik Informatika, STMIK IKMI Cirebon
[3]Sistem Informasi, STMIK IKMI Cirebon
*athharhafizhaluthfi@gmail.com**

Abstract

Imbalanced data is a significant challenge in sentiment analysis, as it often impacts the performance of machine learning models. This study applies the Naïve Bayes algorithm, enhanced with the Synthetic Minority Oversampling Technique (SMOTE), to address class imbalance in user reviews of the by.U application. Using the Knowledge Discovery in Databases (KDD) framework, the research involves data selection, preprocessing (text cleaning, normalization, stemming), transformation using TF-IDF, and train-test data splitting. SMOTE is applied to the training data to improve minority class representation, while Naïve Bayes performs sentiment classification. Model evaluation using cross-validation demonstrates that SMOTE increases accuracy from 84.42% to 85.83%. These results underscore the effectiveness of integrating SMOTE with Naïve Bayes in addressing imbalanced data, offering meaningful insights into user sentiment and aiding the development of improved features for the by.U application.

Keywords: *User Sentiment, Naïve Bayes, SMOTE, Sentiment Analysis, by.U*

1. Introduction

In recent years, advancements in computer science have significantly impacted various aspects of life, including technology, business, and education. Data analysis has become a critical component in leveraging information effectively, particularly in sentiment analysis, where user-generated data is analyzed to identify trends and patterns in feedback [1]. The Naïve Bayes algorithm, known for its efficiency and simplicity, is widely used in sentiment analysis[2]. However, challenges arise when dealing with imbalanced datasets, where minority classes, such as certain sentiment categories, are underrepresented. This issue is evident in applications like by.U, a digital telecommunications platform, where user feedback provides valuable insights for service improvement.

To address this challenge, the Synthetic Minority Oversampling Technique (SMOTE) has emerged as a promising solution. SMOTE improves the representation of minority classes in training data, thus enhancing the performance of classification models [3]. Despite its proven effectiveness, its application in sentiment analysis, particularly with imbalanced data, remains underexplored.

Previous studies have shown the potential of SMOTE in addressing class imbalance. For instance, research on sentiment analysis of Shopee Indonesia users applied SMOTE combined with Tomek Link and achieved promising results, with an accuracy of 80%, precision of 84.1%, and F1-score of 88.1% [4]. Similarly, research by Kasanah et al. [5] demonstrated the effectiveness of SMOTE in improving K-Nearest Neighbor (KNN) classification performance in imbalanced datasets, with a notable increase in accuracy for lower values of K (k=1, k=3). Furthermore, Sutoyo & Fadlurrahman [3] applied SMOTE in classifying TV commercial performance using an Artificial Neural Network (ANN), increasing accuracy from 86.35% to 87.06%. These studies underscore SMOTE's ability to improve classification performance in imbalanced data contexts.

Building on these findings, this study aims to apply SMOTE with the Naïve Bayes algorithm to improve sentiment classification accuracy in the context of by.U user reviews. By addressing class imbalance, the research seeks to enhance the reliability of sentiment analysis and contribute to service improvement and better decision-making in similar applications.

1.1. Naïve Bayes

The Naïve Bayes algorithm is a widely utilized classification method in both Data Mining and Text Mining. This algorithm is based on Bayes' theorem, assuming that all features contribute equally and independently to the determination of a specific class [6].

1.2. Synthetic Minority Over-sampling Technique (SMOTE)

Synthetic Minority Over-sampling Technique is a technique used to generate synthetic data for the minority class, ensuring it becomes proportional to the majority class [7].

1.3. Knowledge Discovery in Database (KDD)

Knowledge Discovery in Databases (KDD) is a method used to extract knowledge from existing databases. The resulting knowledge serves as the foundation for a knowledge base, which can be utilized for decision-making purposes. KDD consists of five stages: data selection, preprocessing, transformation, data mining, and evaluation [8].

2. Research Methodology

2.1. Research Method

This study employs the Knowledge Discovery in Databases (KDD) methodology. The research process or workflow adopted in this study is depicted in Figure 1 below:

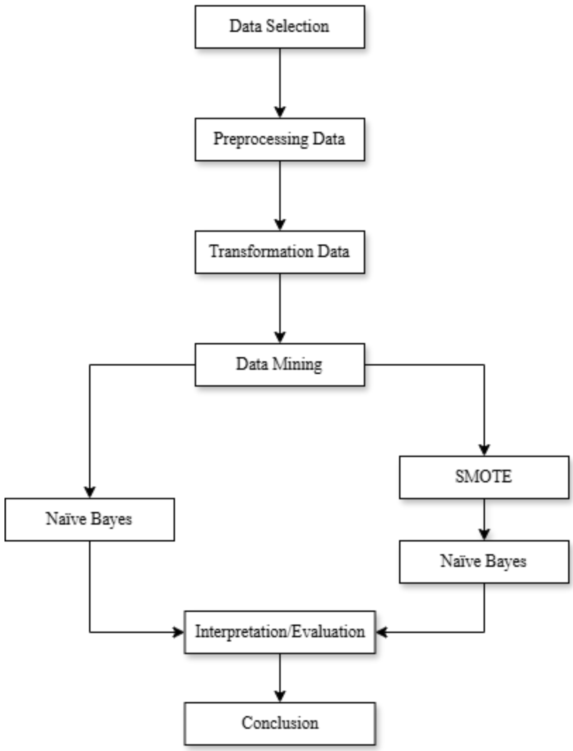

Fig. 1: Research Methodology

3. Result and Discussion

3.1. Data Selection

The data was obtained through web scraping using the Python library "google-play-scraper" to collect user reviews of the by.U app, filtered by the specified date and month and stored in Comma Separated Values (CSV) file format.

Table 1: Sample of Dataset

userName	content	label	thumbsUpCount	reviewCreatedVersion	at
Aanug	Aplikasi Byu sejauh ini sudah sangat baik, dari segi fitur juga sudah lengkap dan saya senang menggunakan Byu karena mudah, namun selama menggunakan aplikasi Byu sangat disayangkan fitur transfer pulsanya hanya di nominal tertentu, saya sarankan kedepannya Aplikasi Byu bisa menyediakan fitur transfer pulsa dengan	3	391	1.54.1	31/05/2024 23:26

Nasir Syan	nominal custom seperti aplikasi MyTelkomsel, atau juga bisa dengan penggunaan kode dial.. Kalo ultah kasih bonus kuota sebulan dong🙏🤣	2	0	1.53.0	31/05/2024 23:01
Ibnu Arya Maulana	Kebanyakan fitur doang, pas dibuka aplikasinya stuck lama di halaman nama depan app doang.	1	0	1.53.0	31/05/2024 22:32
Raon Zieghart	Bagus, tapi bangke sih kenapa sih kecepatan jaringan lemah banget kaya ga sepadan dengan harga kuotanya bangke😴 udah berhari^2 sinyal lemah, kenapa lagi sih?? Ini pake ngga di gunung hp juga mumpuni, gw tinggal juga di kota, kenapa jaringan ilang^{2}n kaya doi yang selalu ngilang🙄 perbaiki lah bang	1	0	NaN	31/05/2024 21:49

3.2. Import Library

After data collection, several libraries are imported to process and analyze the data. These libraries include:

1. Pandas for handling data structures and analysis.
2. Regular expressions (re) and string for cleaning and formatting the text data.
3. NLTK (Natural Language Toolkit) for text tokenization, stemming, and stopword removal.
4. Sastrawi for stemming Indonesian words.
5. Scikit-learn for machine learning tasks, including vectorization, classification, and evaluation metrics.
6. WordCloud for visualizing frequent terms in the dataset.
7. Matplotlib and Seaborn for data visualization.

3.3. Preprocessing

A series of steps or processes are performed on the data before it is used in analysis or modeling. The goal of preprocessing is to clean, normalize, and prepare the data so that it can be processed more effectively by analysis algorithms or machine learning models. This dataset undergoes cleaning, correction, and transformation without altering the validity of the content within the dataset [9]. The text data undergoes preprocessing phase, including text cleaning, case folding, tokenization, stopwords removal, normalization, and stemming using the Sastrawi library for Indonesian language processing. This ensures that the data is formatted and prepared for analysis. An example of the output from the text cleaning phase is shown in Table 2 below.

Table 2: Result of Text Cleaning

Original Data	Cleaned
Aplikasi Byu sejauh ini sudah sangat baik, dari segi fitur juga sudah lengkap dan saya senang menggunakan Byu karena mudah, namun selama menggunakan aplikasi Byu sangat disayangkan fitur transfer pulsanya hanya di nominal tertentu, saya sarankan kedepannya Aplikasi Byu bisa menyediakan fitur transfer pulsa dengan nominal custom seperti aplikasi MyTelkomsel, atau juga bisa dengan penggunaan kode dial..	Aplikasi Byu sejauh ini sudah sangat baik dari segi fitur juga sudah lengkap dan saya senang menggunakan Byu karena mudah namun selama menggunakan aplikasi Byu sangat disayangkan fitur transfer pulsanya hanya di nominal tertentu saya sarankan kedepannya Aplikasi Byu bisa menyediakan fitur transfer pulsa dengan nominal custom seperti aplikasi MyTelkomsel atau juga bisa dengan penggunaan kode dial
Kalo ultah kasih bonus kuota sebulan dong🙏🤣	Kalo ultah kasih bonus kuota sebulan dong
Kebanyakan fitur doang, pas dibuka aplikasinya stuck lama di halaman nama depan app doang.	Kebanyakan fitur doang pas dibuka aplikasinya stuck lama di halaman nama depan app doang
Bagus, tapi bangke sih kenapa sih kecepatan jaringan lemah banget kaya ga sepadan dengan harga kuotanya bangke😴 udah berhari^2 sinyal lemah, kenapa lagi sih?? Ini pake ngga di gunung hp juga mumpuni, gw tinggal juga di kota, kenapa jaringan ilang^{2}n kaya doi yang selalu ngilang🙄 perbaiki lah bang	Bagus tapi bangke sih kenapa sih kecepatan jaringan lemah banget kaya ga sepadan dengan harga kuotanya bangke udah berhari sinyal lemah kenapa lagi sih Ini pake ngga di gunung hp juga mumpuni gw tinggal juga di kota kenapa jaringan ilangn kaya doi yang selalu ngilang perbaiki lah bang

After completing the text cleaning process, the next phase is case folding. Case folding is a text processing technique that converts all characters in a text to a consistent format, typically lowercase [10]. This process aims to reduce variability in the text caused by differences in the use of uppercase and lowercase letters. An example of the output from the case folding phase is shown in Table 3 below.

Table 3: Result of Case Folding

Cleaned	Lowered
Aplikasi Byu sejauh ini sudah sangat baik dari segi fitur juga sudah lengkap dan saya senang menggunakan Byu karena mudah namun selama menggunakan aplikasi Byu sangat disayangkan fitur transfer pulsanya hanya di nominal tertentu saya sarankan kedepannya Aplikasi Byu bisa menyediakan fitur transfer pulsa dengan nominal custom seperti aplikasi MyTelkomsel atau juga bisa dengan penggunaan kode dial	aplikasi byu sejauh ini sudah sangat baik dari segi fitur juga sudah lengkap dan saya senang menggunakan byu karena mudah namun selama menggunakan aplikasi byu sangat disayangkan fitur transfer pulsanya hanya di nominal tertentu saya sarankan kedepannya aplikasi byu bisa menyediakan fitur transfer pulsa dengan nominal custom seperti aplikasi mytelkomsel atau juga bisa dengan penggunaan kode dial
Kalo ultah kasih bonus kuota sebulan dong	kalo ultah kasih bonus kuota sebulan dong
Kebanyakan fitur doang pas dibuka aplikasinya stuck lama di halaman nama depan app doang	kebanyakan fitur doang pas dibuka aplikasinya stuck lama di halaman nama depan app doang
Bagus tapi bangke sih kenapa sih kecepatan jaringan lemah banget kaya ga sepadan dengan harga kuotanya bangke udah berhari sinyal lemah kenapa lagi sih Ini pake ngga di gunung hp juga mumpuni gw tinggal juga di kota kenapa jaringan ilangn kaya doi yang selalu ngilang perbaiki lah bang	bagus tapi bangke sih kenapa sih kecepatan jaringan lemah banget kaya ga sepadan dengan harga kuotanya bangke udah berhari sinyal lemah kenapa lagi sih ini pake ngga di gunung hp juga mumpuni gw tinggal juga di kota kenapa jaringan ilangn kaya doi yang selalu ngilang perbaiki lah bang

After completing the case folding process, the next step is tokenization. Tokenization involves breaking down text or sentences into their smallest units, known as tokens [11]. Tokens can be words, phrases, or other meaningful symbols within the context of linguistics and Natural Language Processing (NLP). The purpose of tokenization is to facilitate text analysis and processing, serving as a foundational step for various NLP tasks, such as sentiment analysis, machine translation, and information extraction. An example of the output from the tokenization phase is shown in Table 4 below.

Table 4: Result of Tokenization

Lowered	Tokenized
aplikasi byu sejauh ini sudah sangat baik dari segi fitur juga sudah lengkap dan saya senang menggunakan byu karena mudah namun selama menggunakan aplikasi byu sangat disayangkan fitur transfer pulsanya hanya di nominal tertentu saya sarankan kedepannya aplikasi byu bisa menyediakan fitur transfer pulsa dengan nominal custom seperti aplikasi mytelkomsel atau juga bisa dengan penggunaan kode dial	['aplikasi','byu','sejauh','ini','sudah','sangat','baik','dari','segi','fitur','juga','sudah','lengkap','dan','saya','senang','menggunakan','byu','karena','mudah','namun','selama','menggunakan','aplikasi','byu','sangat','disayangkan','fitur','transfer','pulsanya','hanya','di','nominal','tertentu','saya','sarankan','kedepannya','aplikasi','byu','bisa','menyediakan','fitur','transfer','pulsa','dengan','nominal','custom','seperti','aplikasi','mytelkomsel','atau','juga','bisa','dengan','penggunaan','kode','dial']
kalo ultah kasih bonus kuota sebulan dong	['kalo','ultah','kasih','bonus','kuota','sebulan','dong']
kebanyakan fitur doang pas dibuka aplikasinya stuck lama di halaman nama depan app doang	['kebanyakan','fitur','doang','pas','dibuka','aplikasinya','stuck','lama','di','halaman','nama','depan','app','doang']
bagus tapi bangke sih kenapa sih kecepatan jaringan lemah banget kaya ga sepadan dengan harga kuotanya bangke udah berhari sinyal lemah kenapa lagi sih ini pake ngga di gunung hp juga mumpuni gw tinggal juga di kota kenapa jaringan ilangn kaya doi yang selalu ngilang perbaiki lah bang	['bagus','tapi','bangke','sih','kenapa','sih','kecepatan','jaringan','lemah','banget','kaya','ga','sepadan','dengan','harga','kuotanya','bangke','udah','berhari','sinyal','lemah','kenapa','lagi','sih','ini','pake','ngga','di','gunung','hp','juga','mumpuni','gw','tinggal','juga','di','kota','kenapa','jaringan','ilangn','kaya','doi','yang','selalu','ngilang','perbaiki','lah','bang']

The next phase after completing tokenization is stop words removal. Stop words removal involves eliminating common words (stop words) that do not contribute to sentiment analysis, such as "yang," "dan," and other functional words [12]. An example of the output from the stop words removal phase is shown in Table 5 below.

Table 5: Result of Stop Words Removal

Tokenized	Stop Words
['aplikasi','byu','sejauh','ini','sudah','sangat','baik','dari','segi','fitur','juga','sudah','lengkap','dan','saya','senang','menggunakan','byu','karena','mudah','namun','selama','menggunakan','aplikasi','byu','sangat','disayangkan','fitur','transfer','pulsanya','hanya','di','nominal','tertentu','saya','sarankan','kedepannya','aplikasi','byu','bisa','menyediakan','fitur','transfer','pulsa','dengan','nominal','custom','seperti','aplikasi','mytelkomsel','atau','juga','bisa','dengan','penggunaan','kode','dial']	aplikasi byu segi fitur lengkap senang byu mudah aplikasi byu disayangkan fitur transfer pulsanya nominal sarankan kedepannya aplikasi byu menyediakan fitur transfer pulsa nominal custom aplikasi mytelkomsel penggunaan kode dial
['kalo','ultah','kasih','bonus','kuota','sebulan','dong']	kalo ultah kasih bonus kuota sebulan
['kebanyakan','fitur','doang','pas','dibuka','aplikasinya','stuck','lama','di','halaman','nama','depan','app','doang']	kebanyakan fitur doang pas dibuka aplikasinya stuck halaman nama app doang
['bagus','tapi','bangke','sih','kenapa','sih','kecepatan','jaringan','lemah','banget','kaya','ga','sepadan','dengan','harga','kuotanya','bangke','udah','berhari','sinyal','lemah','kenapa','lagi','sih','ini','pake','ngga','di','gunung','hp','juga','mumpuni','gw','tinggal','juga','di','kota','kenapa','jaringan','ilangn','kaya','doi','yang','selalu','ngilang','perbaiki','lah','bang']	bagus bangke sih sih kecepatan jaringan lemah banget kaya ga sepadan harga kuotanya bangke udah berhari sinyal lemah sih pake ngga gunung hp mumpuni gw tinggal kota jaringan ilangn kaya doi ngilang perbaiki bang

After the stop words removal phase, the next phase is normalization. Normalization is the process of converting slang or non-standard words into their formal counterparts, making the text more suitable for analysis [7]. In this study, slang normalization was performed using a slang dictionary that contains pairs of slang words and their formal equivalents. This dictionary is stored in a JSON format and serves as a reference to replace slang words with formal ones. An example of the output from the normalization phase is shown in Table 6 below.

Table 6: Result of Normalization

Stop Words	Normalization
aplikasi byu segi fitur lengkap senang byu mudah aplikasi byu disayangkan fitur transfer pulsanya nominal sarankan kedepannya aplikasi byu menyediakan fitur transfer pulsa nominal custom aplikasi mytelkomsel penggunaan kode dial	aplikasi byu segi fitur lengkap senang byu mudah aplikasi byu disayangkan fitur transfer pulsanya nominal sarankan kedepannya aplikasi byu menyediakan fitur transfer pulsa nominal custom aplikasi mytelkomsel penggunaan kode dial
kalo ultah kasih bonus kuota sebulan	kalau ultah kasih bonus kuota sebulan
kebanyakan fitur doang pas dibuka aplikasinya stuck halaman nama app doang	kebanyakan fitur doang pas dibuka aplikasinya stuck halaman nama app doang
bagus bangke sih sih kecepatan jaringan lemah banget kaya ga sepadan harga kuotanya bangke udah berhari sinyal lemah sih pake ngga gunung hp mumpuni gw tinggal kota jaringan ilangn kaya doi ngilang perbaiki bang	bagus bangke sih sih kecepatan jaringan lemah banget kaya ga sepadan harga kuotanya bangke sudah berhari sinyal lemah sih pakai tidak gunung hp mumpuni saya tinggal kota jaringan ilangn kaya doi ngilang perbaiki bang

The final phase in preprocessing is stemming. Stemming is the process of converting words into their root forms to facilitate further analysis, so variations of words with the same root, such as "berjalan" and "berjalanlah," are considered as a single entity [13]. An example of the output from the stemming phase is shown in Table 7 below.

Table 7: Result of Stemming

Normalization	Stemming
aplikasi byu segi fitur lengkap senang byu mudah aplikasi byu disayangkan fitur transfer pulsanya nominal sarankan kedepannya aplikasi byu menyediakan fitur transfer pulsa nominal custom aplikasi mytelkomsel penggunaan kode dial	aplikasi byu segi fitur lengkap senang byu mudah aplikasi byu sayang fitur transfer pulsa nominal saran depan aplikasi byu sedia fitur transfer pulsa nominal custom aplikasi mytelkomsel guna kode dial
kalau ultah kasih bonus kuota sebulan	kalau ultah kasih bonus kuota bulan
kebanyakan fitur doang pas dibuka aplikasinya stuck halaman nama app doang	banyak fitur doang pas buka aplikasi stuck halaman nama app doang
bagus bangke sih sih kecepatan jaringan lemah banget kaya ga sepadan harga kuotanya bangke sudah berhari sinyal lemah sih pakai tidak gunung hp mumpuni saya tinggal kota jaringan ilangn kaya doi ngilang perbaiki bang	bagus bangke sih sih cepat jaring lemah banget kaya ga padan harga kuota bangke sudah hari sinyal lemah sih pakai tidak gunung hp mumpuni saya tinggal kota jaring ilangn kaya doi ngilang baik bang

3.4. Transformation

After preprocessing, the data is transformed into a numerical format suitable for machine learning models using TF-IDF. This technique assigns weights to words based on their frequency in a document and their rarity across the dataset [14]. Subsequently, the data is split into training and testing sets, with an 80:20 ratio, using the train-test split technique. After performing TF-IDF, the next step is to apply the SMOTE technique to balance the imbalanced data. The outcome of the SMOTE process is shown in the Figure 2 below.

	count
label	
3	974
1	974
2	974

dtype: int64

Fig. 2: Train set after SMOTE

After applying SMOTE, synthetic data points are generated for the minority class, balancing the dataset by increasing it's representation. This helps the model to learn patterns from both classes more equally, reducing bias towards the majority class. The balanced data is then used to train machine learning models, improving the model's performance, especially for the minority class, and ensuring better generalization with cross-validation.

3.5. Data Mining

At this stage, cross-validation is utilized to fine-tune the hyperparameters of the Naïve Bayes models through a 10-fold cross-validation method. The cross_val_score function from the scikit-learn library is employed to ensure consistent model assessment and prevent overfitting. The outcome of the parameter search for the Naïve Bayes model without SMOTE is displayed in the Figure 3 below.

```
Best parameters found:  {'alpha': 0.1, 'fit_prior': True}
Best cross-validation accuracy: 84.42%
```

Fig. 3: GridSearch result Naïve Bayes Model without SMOTE

The image shows the results of the parameter search for the Naïve Bayes model before applying the SMOTE technique. The optimal parameters identified were alpha set to 0.1 and fit_prior set to True. The best cross-validation accuracy achieved at this stage was 84.42%. This performance reflects the results from the Naïve Bayes model on the original dataset, which may be affected by class imbalance. After applying SMOTE, which generates synthetic samples for the minority class, the model's performance is expected to improve as the class distribution becomes more balanced. This will be shown in subsequent results. The outcome of the parameter search for the Naïve Bayes model with SMOTE is displayed in the Figure 4 below.

```
Best parameters found:  {'alpha': 0.1, 'fit_prior': True}
Best cross-validation accuracy: 85.83%
```

Fig. 4: GridSearch result Naïve Bayes Model with SMOTE

Compared to the results obtained without using SMOTE, where the model achieved an accuracy of 84.42%, applying SMOTE with GridSearchCV improved the model's accuracy to 85.83%. This demonstrates that the use of SMOTE successfully balanced the class distribution in the training data, enabling the model to be more sensitive to the minority class and resulting in better performance compared to a model trained solely on the imbalanced original data.

3.6. Evaluation

The final stage of the KDD process is evaluation of the classification model's results. The model's performance is assessed using metrics such as accuracy, precision, recall, and F1-score. Cross-validation is employed to divide the data into several folds for training and testing, reducing bias and providing a more reliable estimate of model performance [15]. A confusion matrix is also used to give a detailed view of the model's performance, showing the correct and incorrect predictions for each class. This matrix helps calculate accuracy, precision, recall, and F1-score. Additionally, a word cloud visualization is created to display the most frequent words in each sentiment category (positive, negative, and neutral). The results of the classification report without SMOTE, including the metrics of precision, recall, F1-score, and support for each class, can be seen in the Table 8 below.

Table 8: Classification Report without SMOTE

Label	Precision	Recall	F1-Score	Accuracy	Support
1	80.00%	96.65%	87.54%	84.08%	1225
2	0%	0%	0%	84.08%	128
3	91.18%	79.32%	84.84%	84.08%	977

According to the Table 8, the results show the performance of the model without using the SMOTE technique for each sentiment label: Label 1 (Negative), Label 2 (Neutral), and Label 3 (Positive). For Label 1 (Negative), the precision of 80.00% indicates that 80% of the negative sentiment predictions were correct. The recall of 96.65% means that almost all actual negative sentiment data were successfully detected by the model. With an F1-Score of 87.54%, the model demonstrates a good balance between precision and recall for negative sentiment. The overall accuracy remains at 84.07%.

However, for Label 2 (Neutral), the model failed to predict neutral sentiment at all, with precision, recall, and F1-Score all at 0%. This indicates that the model is not sensitive to the neutral class, likely due to the limited amount of data in this class compared to others (data imbalance). Nonetheless, accuracy remains at 84.07%, as the model tends to ignore the neutral class and more frequently predicts the majority class.

For Label 3 (Positive), precision of 91.17% shows that most of the positive sentiment predictions were correct, while recall of 79.32% indicates that the model was able to identify around 79% of the actual positive data. The F1-Score of 84.83% reflects a good balance between precision and recall for positive sentiment.

Overall, this table demonstrates that the model is highly biased towards the majority classes (negative and positive) but fails to recognize the minority class (neutral). Although the accuracy appears high (84.07%), it does not fully reflect the model's performance, particularly for the minority class. The application of techniques like SMOTE is necessary to balance the data and improve the model's performance on the neutral class. The Confusion Matrix produced by the Naïve Bayes model without SMOTE for the classification applied to the test (training) set is displayed in Figure 5 below.

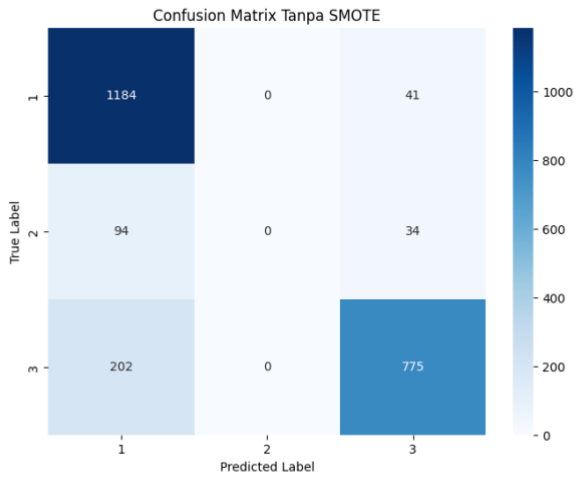

Fig. 5: Confusion Matrix Naive Bayes Model without SMOTE

Figure 5 displays the confusion matrix visualization for the Naïve Bayes model without the SMOTE technique. The model made 1184 correct predictions for the negative class (1), 775 for the positive class (3), and 0 for the neutral class (2). However, there were also some misclassifications, such as 41 incorrect predictions for the negative class (1) (0 were classified as neutral and 41 as positive), and 202 incorrect predictions for the positive class (202 were classified as negative and none as neutral). The model did not predict any neutral class instances due to the very small amount of neutral data compared to the other classes (data imbalance). These results indicate that the Naïve Bayes model without SMOTE performs well in predicting the negative and positive classes but struggles with the neutral class. Based on this confusion matrix visualization, it can be concluded:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1} - \text{Score} = 2 \times \frac{\text{Precision} \ \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Known:
1. TP = True Positive
2. FP = False Positive
3. FN = False Negative

1. Precision Matrix

Therefore, the Precision of the Positive class (3):
TP = 775
FP = 75

$$\text{Precision} = \frac{775}{775 + 75} = 0.911$$

Therefore, the Precision of the Negative class (1):
TP = 1184
FP = 296

$$\text{Precision} = \frac{1184}{1184 + 296} = 0.8$$

Therefore, the Precision of the Neutral class (2):
TP = 0
FP = 0

$$\text{Precision} = \frac{0}{0 + 0} = 0$$

2. Recall Matrix

Therefore, the Recall of the Positive class (3):
TP = 775
FN = 202

$$\text{Recall} = \frac{775}{775 + 202} = 0.793$$

Therefore, the Recall of the Negative class (1):

TP = 1184
FN = 41

$$\text{Recall} = \frac{1184}{1184 + 41} = 0.966$$

Therefore, the Recall of the Neutral class (2):
TP = 0
FN = 0

$$\text{Recall} = \frac{0}{0 + 0} = 0$$

3. F1-Score Matrix
Therefore, the F1-Score of the Positive class (3):
Precision = 0.911
Recall = 0.793

$$\text{F1} - \text{Score} = 2 \times \frac{0.911 \times 0.793}{0.911 + 0.793} = 0.848$$

Therefore, the F1-Score of the Negative class (1):
Precision = 0.8
Recall = 0.966

$$\text{F1} - \text{Score} = 2 \times \frac{0.8 \times 0.966}{0.8 + 0.966} = 0.875$$

Therefore, the F1-Score of the Neutral class (2):
Precision = 0
Recall = 0

$$\text{F1} - \text{Score} = 2 \times \frac{0 \times 0}{0 + 0} = 0$$

The results of the classification report with SMOTE, including the metrics of precision, recall, F1-score, and support for each class, can be seen in the Table 9 below.

Table 9: Classification Report with SMOTE

Label	Precision	Recall	F1-Score	Accuracy	Support
1	81.34%	92.20%	86.43%	85.49%	974
2	84.24%	85.63%	84.93%	85.49%	974
3	92.51%	78.64%	85.02%	85.49%	974

Table 9 shows the performance of the model after using the SMOTE technique for each sentiment label: Label 1 (Negative), Label 2 (Neutral), and Label 3 (Positive). For Label 1 (Negative), precision reaches 81.34%, meaning that more than 81% of the negative sentiment predictions are correct. The recall of 92.19% indicates that the model is able to detect most of the actual negative sentiment data. With an F1-Score of 86.42%, the model demonstrates a good balance between precision and recall for this label. The overall accuracy is 85.48%, reflecting that the model provides more consistent predictions.

For Label 2 (Neutral), precision of 84.24% suggests that most neutral sentiment predictions are correct, while recall of 85.62% indicates that the model effectively detects actual neutral sentiment data. The F1-Score of 84.92% shows a balanced and reliable performance for neutral sentiment, which was previously a weakness in the model without SMOTE.

For Label 3 (Positive), precision stands at 92.51%, indicating that nearly all positive sentiment predictions are correct. However, recall of 78.64% shows that the model still misses some of the actual positive sentiment data. The F1-Score of 85.01% reflects that the performance on this label remains strong, even though recall is slightly lower than precision.

Overall, the application of SMOTE has improved the model's performance for all sentiment labels, including the neutral label, which was previously not well detected. The consistent accuracy of 85.48% across all labels reflects the model’s improvement in handling data imbalance, resulting in fairer and more accurate predictions for negative, neutral, and positive sentiments. The Confusion Matrix produced by the Naïve Bayes model with SMOTE for the classification applied to the test (training) set is displayed in Figure 6 below.

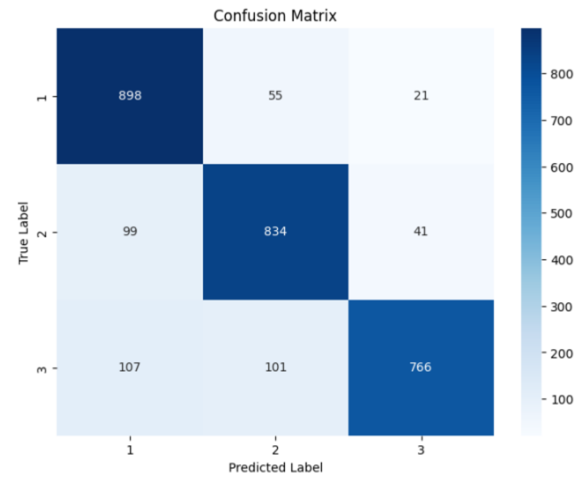

Fig. 6: Confusion Matrix Naive Bayes Model with SMOTE

Figure 6 displays the visualization of the confusion matrix for the Naïve Bayes model with SMOTE technique shows the following results: the model correctly predicted 898 instances for the negative class (1), 766 for the positive class (3), and 834 for the neutral class (2). However, there were also several misclassifications: 76 predictions for the negative class (1) were incorrect (55 were classified as neutral and 21 as positive), 208 predictions for the positive class were wrong (107 were classified as negative and 101 as neutral), and 140 predictions for the neutral class were incorrect (99 were classified as negative and 41 as positive). Based on this confusion matrix visualization, it can be concluded:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1} - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Known:

4. TP = True Positive
5. FP = False Positive
6. FN = False Negative

4. Precision Matrix

Therefore, the Precision of the Positive class (3):
TP = 766
FP = 62

$$\text{Precision} = \frac{766}{766 + 62} = 0.925$$

Therefore, the Precision of the Negative class (1):
TP = 898
FP = 206

$$\text{Precision} = \frac{898}{898 + 206} = 0.813$$

Therefore, the Precision of the Neutral class (2):
TP = 834
FP = 156

$$\text{Precision} = \frac{834}{834 + 156} = 0.842$$

5. Recall Matrix

Therefore, the Recall of the Positive class (3):
TP = 766
FN = 208

$$\text{Recall} = \frac{766}{766 + 208} = 0.786$$

Therefore, the Recall of the Negative class (1):
TP = 898

FN = 76

$$\text{Recall} = \frac{898}{898 + 76} = 0.921$$

Therefore, the Recall of the Neutral class (2):
TP = 834
FN = 140

$$\text{Recall} = \frac{834}{834 + 140} = 0.856$$

6. F1-Score Matrix
Therefore, the F1-Score of the Positive class (3):
Precision = 0.925
Recall = 0.786

$$\text{F1} - \text{Score} = 2 \times \frac{0.925 \times 0.786}{0.925 + 0.786} = 0.85$$

Therefore, the F1-Score of the Negative class (1):
Precision = 0.813
Recall = 0.921

$$\text{F1} - \text{Score} = 2 \times \frac{0.813 \times 0.921}{0.813 + 0.921} = 0.864$$

Therefore, the F1-Score of the Neutral class (2):
Precision = 0.842
Recall = 0.856

$$\text{F1} - \text{Score} = 2 \times \frac{0.842 \times 0.856}{0.842 + 0.856} = 0.849$$

The results of the data preprocessing are displayed in the form of a word cloud to analyze the most commonly used words in user reviews of the by.U app. This visualization includes all the data, covering positive, negative, and neutral reviews, which are analyzed separately, as shown below.

Fig. 7: Positive Word Cloud

Fig. 8: Negative Word Cloud

Fig. 9: Neutral Word Cloud

Based on the research results, the accuracy of the model without using SMOTE is 84.07%, while with the application of SMOTE, the accuracy increased to 85.48%. Although this improvement is not significant, it indicates that the SMOTE technique contributes positively to addressing class imbalance and enhancing the model's accuracy in sentiment classification.

In comparison, previous studies that used the Naïve Bayes algorithm without SMOTE on datasets with significant class imbalance tended to experience a decline in classification performance [5]. The application of SMOTE in this study strengthens the findings of other research, such as the study conducted in reference [3], which showed an accuracy increase from 86.35% to 87.06% after SMOTE was applied to television ad classification. This study supports the conclusion that SMOTE is effective in addressing data imbalance issues in sentiment analysis, particularly for the by.U app.

Additionally, the study in reference [4], which combined SMOTE with machine learning algorithms for sentiment analysis of the Shopee app, also showed improved model performance on imbalanced data. The results of this research further reinforce the hypothesis that applying SMOTE with the Naïve Bayes algorithm can be a promising solution to enhance model performance on imbalanced datasets, which are common in user review sentiment analysis.

Overall, the findings of this study demonstrate that combining the SMOTE technique with the Naïve Bayes algorithm can improve the accuracy of sentiment classification models. These findings have significant implications for the development of more robust and reliable classification methods for sentiment analysis, as well as contributing to the development of systems that can provide more accurate results in class-imbalanced scenarios. Therefore, the results of this study can serve as a reference for future research in machine learning and data mining focused on handling imbalanced data.

4. Conclusion

The SMOTE technique successfully addresses the data imbalance issue in the user sentiment dataset. By adding synthetic samples to the minority classes, namely negative and neutral sentiments, the data distribution becomes more balanced. This enables the model to learn sentiment patterns more effectively, which were previously difficult to identify due to the dominance of majority sentiment data. The application of SMOTE significantly contributes to the model's ability to classify sentiment in the minority classes. SMOTE implementation improved the model's accuracy from 84.07% to 85.48%. Although the improvement is not highly significant, the results demonstrate that SMOTE can enhance model performance on imbalanced datasets. In addition to accuracy, other evaluation metrics such as precision, recall, and F1-score also showed consistent improvements. These results align with previous research findings, which suggest that SMOTE can improve the accuracy of classification algorithms on imbalanced datasets.

References

[1] F. Sidik, I. Suhada, A. H. Anwar, and F. N. Hasan, "Analisis Sentimen Terhadap Pembelajaran Daring Dengan Algoritma Naive Bayes Classifier," *J. Linguist. Komputasional*, vol. 5, no. 1, pp. 34–43, 2022, doi: 10.26418/jlk.v5i1.79.

[2] I. Wickramasinghe and H. Kalutarage, "Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation," *Soft Comput.*, vol. 25, no. 3, pp. 2277–2293, 2021, doi: 10.1007/s00500-020-05297-6.

[3] E. Sutoyo and M. A. Fadlurrahman, "Penerapan SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Television Advertisement Performance Rating Menggunakan Artificial Neural Network," *J. Edukasi dan Penelit. Inform.*, vol. 6, no. 3, pp. 379–385, 2020, doi: 10.26418/jp.v6i3.42896.

[4] H. Utami, "Analisis Sentimen dari Aplikasi Shopee Indonesia Menggunakan Metode Recurrent Neural Network," *Indones. J. Appl. Stat.*, vol. 5, no. 1, pp. 31–38, 2022, doi: 10.13057/ijas.v5i1.56825.

[5] A. N. Kasanah, M. Muladi, and U. Pujianto, "Penerapan Teknik SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Objektivitas Berita Online Menggunakan Algoritma KNN," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 3, no. 2, pp. 196–201, 2019, doi: 10.29207/resti.v3i2.945.

[6] D. Darwis, N. Siskawati, and Z. Abidin, "Penerapan Algoritma Naive Bayes Untuk Analisis Sentimen Review Data Twitter Bmkg Nasional," *J. Tekno Kompak*, vol. 15, no. 1, pp. 131–145, 2021, doi: 10.33365/jtk.v15i1.744.

[7] J. W. Iskandar and Y. Nataliani, "Perbandingan Naïve Bayes, SVM, dan k-NN untuk Analisis Sentimen Gadget Berbasis Aspek," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 6, pp. 1120–1126, 2021, doi: 10.29207/resti.v5i6.3588.

[8] I. S. Hidayat, S. Defit, and G. W. Nurcahyo, "Simulasi dalam Optimalisasi Pengadaan Barang menggunakan Metode K-Mean Clustering," *J. Sistim Inf. dan Teknol.*, vol. 3, no. 4, pp. 281–286, 2021, doi: 10.37034/jsisfotek.v3i4.79.

[9] F. Putra, H. F. Tahiyat, R. M. Ihsan, R. Rahmaddeni, and L. Efrizoni, "Penerapan Algoritma K-Nearest Neighbor Menggunakan Wrapper Sebagai Preprocessing untuk Penentuan Keterangan Berat Badan Manusia," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 1, pp. 273–281, 2024, doi: 10.57152/malcom.v4i1.1085.

[10] L. A. Andika, P. A. N. Azizah, and R. Respatiwulan, "Analisis Sentimen Masyarakat terhadap Hasil Quick Count Pemilihan Presiden Indonesia 2019 pada Media Sosial Twitter Menggunakan Metode Naive Bayes Classifier," *Indones. J. Appl. Stat.*, vol. 2, no. 1, p. 34, 2019, doi: 10.13057/ijas.v2i1.29998.

[11] W. Yulita, E. D. Nugroho, and M. H. Algifari, "Analisis Sentimen Terhadap Opini Masyarakat Tentang Vaksin Covid-19 Menggunakan Algoritma Naïve Bayes Classifier," *J. Data Min. dan Sist. Inf.*, vol. 2, no. 2, pp. 1–9, 2021, doi: 10.33365/jdmsi.v2i2.1344.
[12] A. Karimah, G. Dwilestari, and M. Mulyawan, "Analisis Sentimen Komentar Video Mobil Listrik Di Platform Youtube Dengan Metode Naive Bayes," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 1, pp. 767–737, 2024, doi: 10.36040/jati.v8i1.8373.
[13] E. Fitri, "Analisis Sentimen Terhadap Aplikasi Ruangguru Menggunakan Algoritma Naive Bayes, Random Forest Dan Support Vector Machine," *J. Transform.*, vol. 18, no. 1, pp. 71–80, 2020, doi: 10.26623/transformatika.v18i1.2317.
[14] A. I. Tanggraeni and M. N. N. Sitokdana, "Analisis Sentimen Aplikasi E-Government pada Google Play Menggunakan Algoritma Naïve Bayes," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 2, pp. 785–795, 2022, doi: 10.35957/jatisi.v9i2.1835.
[15] C. A. Ramezan, T. A. Warner, and A. E. Maxwell, "Evaluation of sampling and cross-validation tuning strategies for regional-scale machine learning classification," *Remote Sens.*, vol. 11, no. 2, pp. 1–21, 2019, doi: 10.3390/rs11020185.