Comparison of Naive Bayes, Decision Trees and SVM Algorithms for Sentiment Classification of JMO Applications

Anas Nasrulloh[*1], Muhamad Yusuf[2], Ibnu Mas'ud[3], Tubagus Toifur[4], Aolia Ikhwanudin[5], Agianto Syamhalim[6]
[1,5,6] Information Systems Study Program, Faculty of Computer Science, South Tangerang Institute of Technology, Tangerang
[3,4] Information Technology Study Program, Faculty of Computer Science, South Tangerang Institute of Technology, Tangerang
[2] Informatics Study Program, Faculty of Computer Science, South Tangerang Institute of Technology, Tangerang
E-mail: [*1]**anas@itts.ac.id**, [2]yusuf@itts.ac.id, [3]ibnu@itts.ac.id, [4]aikwanudin@itts.ac.id, [5]tubagus@itts.ac.id, [6]agianto@itts.ac.id

Abstract

In this study, the researchers found that SVM achieved a precision of 0.75 for negative sentiment and 0.93 for positive sentiment, with recalls of 0.86 and 0.94, and f1-scores of 0.80 and 0.94, and an overall accuracy of 0.88. Naive Bayes showed similar results with a precision of 0.74 for negative and 0.93 for positive, recalls of 0.87 and 0.94, f1-scores of 0.80 and 0.94, and an accuracy of 0.88. Meanwhile, Decision Tree had the lowest precision for negative (0.71) and positive (0.91) sentiment, with recalls of 0.73 and 0.93, f1-scores of 0.72 and 0.92, and an accuracy of 0.85. These findings suggest that SVM and Naive Bayes offer excellent performance in sentiment classification, while Decision Tree, while still effective, performed slightly lower. These results provide valuable guidance in selecting the right algorithm for sentiment analysis on app data. This study compares the effectiveness of three machine learning algorithms—Naive Bayes, Decision Trees, and Support Vector Machine (SVM)—in sentiment classification of JMO apps using review data taken from Google Play Store via web scraping and processed with a Python application. The evaluation is done based on precision, recall, f1-score, and accuracy metrics.

Keywords *— Decision Tree, JMO Application, Naive Bayes, SVM*

1. INTRODUCTION

In today's digital era, sentiment analysis plays a vital role in understanding how apps are received by users. With the increasing amount of review data available on platforms like the Google Play Store, sentiment classification techniques have become a crucial tool for interpreting user opinions.

Some previous relevant studies, including the following: In the study entitled "Sentiment Analysis of MyPertamina Application Reviews on Google Play Store Using the NBC Algorithm", using Google Colab and the Python programming language to analyze 5722 MyPertamina application reviews, which were divided into 80% training data and 20% test data. The algorithm used was the Naïve Bayes Classifier (NBC), resulting in an accuracy of

87%, precision of 86%, recall of 90%, and f1-score of 87% [1]. In the study entitled "Sentiment Analysis of Dana Application Reviews with the Random Forest Method", using the Random Forest method, obtained an accuracy of 84%, precision of 84%, recall of 84%, and f1-score of 84% with a data division of 80% for training and 20% for testing. The tree depth is 65 and the number of trees is 400 [2]. In the study entitled "Sentiment Analysis on Kredivo Application Reviews with SVM and NBC Algorithms", comparing the Support Vector Machine (SVM) and Naïve Bayes Classifier (NBC) algorithms. SVM provides an accuracy of 83.3% with a precision of 77% for the positive class and 87% for the negative class. The recall is 89% for the positive class and 73% for the negative class. While NBC produces an accuracy of 80.8% with a precision of 81% for the positive class and 87% for the negative class, and a recall of 88% for the positive class and 79% for the negative class [3]. In the study entitled "Sentiment Analysis of Brimo Applications on User Reviews on Google Play Using the Naïve Bayes Algorithm", using review data taken from August 2022 to January 2023, as many as 1550 reviews. Analysis using Naïve Bayes showed an accuracy of 84.52%, precision of 82.51%, and recall of 87.62%. Positive reviews reached 1012 and negative reviews 894 [4]. In the study entitled "Sentiment Analysis of Ajaib Application User Reviews Using the Naïve Bayes Method", using Naïve Bayes on 500 review data, the results were 44.2% positive reviews and 55.8% negative reviews. The accuracy obtained was 74.44% [5]. In the study entitled "Sentiment analysis of the TikTok application using the naïve bayes algorithm and support vector machine", testing Naïve Bayes and SVM on 2000 review data. Naïve Bayes produced an accuracy of 79%, while SVM achieved an accuracy of 84% [6]. In the study entitled "Sentiment Analysis on Amazon Shopping Application Reviews on Google Play Store Using Naïve Bayes Classifier", using various Naïve Bayes algorithms, the average accuracy was 82.15% with a precision of 72.25%, a recall of 83.49%, and an f1-score of 77.41%. Multinomial Naïve Bayes provided the highest accuracy of 86.74% [7].

This study aims to compare the effectiveness of three machine learning algorithms in performing sentiment classification of the JMO app, namely Naive Bayes, Decision Trees, and Support Vector Machine (SVM). The app review data was obtained through web scraping and processed using a Python application for further evaluation. The results of the study showed that the SVM algorithm delivered solid performance with a precision of 0.75 for negative sentiment and 0.93 for positive sentiment, as well as comparable recall and f1-score, resulting in an overall accuracy of 0.88. Naive Bayes showed almost identical results to SVM, with very similar precision, recall, and f1-score, also achieving an accuracy of 0.88. Meanwhile, Decision Tree showed slightly lower performance with precision for negative sentiment of 0.71 and positive sentiment of 0.91, and recall and f1-score were also lower than the other two algorithms, with an overall accuracy of 0.85. These findings provide valuable insights into the effectiveness of different algorithms in sentiment classification and can help in selecting the most suitable method for practical applications in user opinion analysis.

2. RESEARCH METHOD

Data collected from user reviews on the JMO application on Google Play In research or analysis involving application review data, such as JMO application reviews taken from the Google Play Store on Aug 19, 2024, as many as 100,000 reviews, which were from July 17, 2023 to Aug 18, 2024, the data structure obtained from the scraping process has the following

columns. Each column stores specific information that is important for sentiment analysis and further study. Here is an explanation of each column:

- *reviewId* is a unique identifier for each review, ensuring that each piece of feedback can be clearly referenced.
- *userName* records the name of the user who left the review, while
- *userImage* provides the URL or path to the user's profile picture, adding a personal dimension to the review.
- *content* is the main text of the review, which is the primary focus of sentiment analysis and text processing, allowing for assessment of sentiment—whether positive, negative, or neutral—and extraction of other important information.
- *score* is a numeric rating given by users, usually on a scale of 1 to 5, and is used to determine sentiment and understand the distribution of app ratings.
- *thumbsUpCount* shows the number of "likes" or "thumbs ups" a review received, indicating how helpful or liked it was by other users.
- *reviewCreatedVersion* records the version of the app when the review was written,
- *at* shows the date and time when the review was posted.
- *replyContent* is the developer or app team's response to the review, with
- *repliedAt* indicating when the response was given
- *appVersion* records the version of the app installed by the user when writing the review, providing additional context for analyzing reviews based on different app versions.

In this study, the steps used are as shown in Figure 2.1 below:

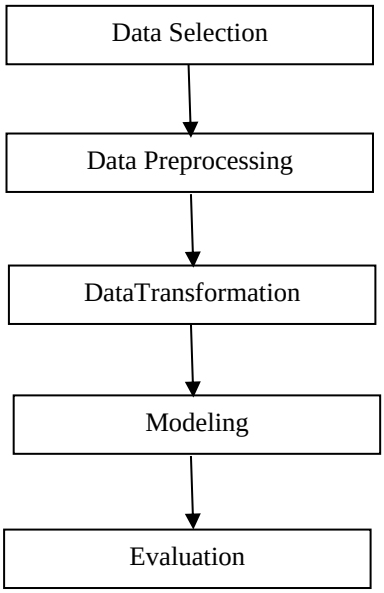


Figure 1. Research Steps

From Figure 1 above, it can be explained as follows:

1. Data Selection

In the data selection stage, the first step is to import review data from a CSV file into a format that can be analyzed, namely DataFrame. This process is done using the pd.read_csv(' jamsostek_mobile_reviews.csv') command, which converts a CSV file containing review data into a DataFrame. This conversion is important because DataFrame provides a structure that makes it easier to process, analyze, and manipulate review data. By using DataFrame, we can explore data more efficiently and prepare it

for a more in-depth analysis stage. DataFrame allows us to manage and filter information in a more structured way, so that the analysis process becomes more organized and effective. This initial step is an important foundation for starting data analysis, because the DataFrame structure makes it easier to process and interpret review data that will be analyzed further.

2. Data Pre-processing

In the data pre-processing stage, various steps are taken to prepare the review data so that it is ready for in-depth analysis. The first step is to download a list of stopwords, which are common words that do not provide much information, which will be used to clean the text. This process is done using the nltk.download('stopwords') command. Next, we need to examine the first few rows of the review data to understand the structure and content of the data. This is done using the print(reviews_df.head()) command, which provides an overview of the format and content of the data.

After that, the sentiment labels contained in the data need to be transformed into simpler categories, such as 'positive', 'neutral', or 'negative', based on the scores given in the reviews. To do this, we define the simplify_sentiment function which is then applied to the 'score' column to simplify the sentiment labels. The next process is text pre-processing, which involves cleaning the text by removing non-alphabetic characters, numbers, and stopwords. This aims to improve the quality of the data and facilitate analysis. The preprocess_text function is applied to format the review text as needed.

Finally, once the text cleaning is complete, we need to re-examine the processed data to ensure that it has been done correctly. This is also done with the print(reviews_df.head()) command, which allows us to see the final result of the data cleaning process. These steps together ensure that the review data is ready for further analysis.

3. Data Transformation

In the data transformation stage, the first step is to split the dataset into two parts: training data and test data. The purpose of this split is to train the model with one subset of the data and test its performance with a different subset of the data. This process is done using the train_test_split function, which splits the review data and sentiment labels into two sets, where 80% is used to train the model and the remaining 20% for testing. This process ensures that the model can be objectively assessed based on data that has never been seen before.

The next step is to build a pipeline that integrates the various steps of data processing and model training in one workflow. This pipeline simplifies the analysis process by unifying several stages, from feature extraction to model training. In this case, the pipeline is defined using CountVectorizer to convert the review text into a numeric feature representation, TfidfTransformer to calculate the TF-IDF of the numeric features, and DecisionTreeClassifier or NaiveBayesClassifier or SVMClassifier as the classification model. By building this pipeline, the entire process from data processing to model training can be done efficiently and in a structured manner.

4. Modeling

In the modeling stage, the first step is to train the model with training data. The main goal of this step is for the classification model to learn patterns from the data and improve its ability to produce accurate predictions. The training process is carried out by running the pipeline.fit(X_train, y_train) command, which teaches the model based on the given feature and label data. After the model is trained, the next step is to test its ability by predicting labels on the test data. This is done by running the pipeline.predict(X_test) command, which generates predictions for test data that the model has never seen before. In this way, the model can evaluate its ability to make accurate predictions on data that was not included in the training process, providing an idea of how well the model can be applied to new and untrained data. These steps are important to ensure that the model can generalize and provide reliable results.

5. Evaluation

In the evaluation stage, we evaluate the performance of the classification model using several important metrics. Evaluation aims to evaluate the extent to which the model can produce accurate predictions. The first step in this process is to import the confusion_matrix function from the scikit-learn library to generate a confusion matrix. A confusion matrix is a tool that shows how the model's prediction results compare to the actual labels of the test data. By calculating the confusion matrix, we can see the number of correct and incorrect predictions for each category, based on the labels specified in pipeline.classes_. This matrix is then displayed to provide a clear picture of how well the model is at classifying data into the correct categories.

The next step in this process is to calculate the accuracy of the model, which is done using the accuracy_score(y_test, y_pred) function. This function provides the percentage of correct predictions compared to the total number of test data. After that, a classification report is generated using the classification_report(y_test, y_pred) function, which presents additional metrics such as precision, recall, and F1 score for each class. By calculating the accuracy and compiling a classification report, we can comprehensively evaluate the effectiveness of the model. This evaluation provides a clear picture of how well the model can identify the correct class and how well the model handles different types of errors. Both steps are crucial to ensure that the model not only predicts accurately but also performs a thorough analysis of its performance on each class in the test data.

3. RESEARCH RESULTS AND DISCUSSION

In a study entitled "Comparative Analysis of Naive Bayes, SVM, and Decision Tree Algorithms in Classifying Sentiment of Dana Application Reviews from Google Play Store Using Python," an evaluation of the performance of three classification algorithms was conducted, namely Support Vector Machine (SVM), Naive Bayes, and Decision Tree. This analysis aims to understand the effectiveness of each algorithm in classifying application review sentiment into negative and positive categories. The evaluation results show significant differences in performance between the three algorithms, which can be described as follows:

The resulting confusion matrix for Naive Bayes, Decision Trees dan SVM can be seen in table 1 as follows:

Table 1. Confusion Matrix

Naïve Bayes			**Decision Trees**			**SVM**		
3823	0	560	3220	212	951	3788	1	594
475	0	432	407	49	451	466	0	441
859	0	13851	902	141	13667	828	1	13881

Table 1. contains the following information:

- Naïve Bayes:
  - Positive: 3823 correct predictions, 0 incorrect prediction as negative, and 560 incorrect predictions as neutral.
  - Negative: 475 incorrect predictions as positive, 0 correct predictions, and 432 incorrect predictions as neutral.
  - Neutral: 859 incorrect predictions as negative, 0 incorrect prediction as positive, and 13851 correct predictions.
  - This confusion matrix indicates that the Naive Bayes model tends to be very accurate in identifying neutral sentiment, but has challenges in distinguishing between positive and negative sentiment.

- Decision Trees:
  - Positive: 3220 correct predictions, 212 incorrect prediction as negative, and 951 incorrect predictions as neutral.
  - Negative: 407 incorrect predictions as positive, 49 correct predictions, and 451 incorrect predictions as neutral.
  - Neutral: 902 incorrect predictions as negative, 141 incorrect prediction as positive, and 13667 correct predictions.
  - This confusion matrix indicates that the Decision Trees model tends to be very accurate in identifying neutral sentiment, but has challenges in distinguishing between positive and negative sentiment.

- SVM:
  - Positive: 3788 correct predictions, 1 incorrect prediction as negative, and 594 incorrect predictions as neutral.
  - Negative: 466 incorrect predictions as positive, 0 correct predictions, and 441 incorrect predictions as neutral.
  - Neutral: 828 incorrect predictions as negative, 1 incorrect prediction as positive, and 13881 correct predictions.
  - This confusion matrix indicates that the SVM model tends to be very accurate in identifying neutral sentiment, but has challenges in distinguishing between positive and negative sentiment.

1. Naïve Bayes

Evaluation Metrics

Evaluation metrics are obtained from calculations using the confusion matrix value from table 1, the calculation is as follows:

- Precision

Precision measures how accurate the model is in classifying reviews into the correct category. This shows the proportion of reviews that truly match the category predicted by the model.

- Precision for Negative Sentiment

  Of the total reviews predicted as negative, 74% are truly negative. The calculation is:
  Precision (Negative)=3823 (True-negative reviews))/(3823+475+859)≈0.74
  This means that 74% of all reviews predicted as negative are indeed negative.
- Precision for Positive Sentiment

  Of the total reviews predicted as positive, 93% are truly positive. The calculation is:
  Precision (Positive)=13851 (True-positive reviews)/(13851+560+432)≈0.93
  This means that 93% of all reviews predicted to be positive are indeed positive.

- Recall

Recall measures how well the model finds all reviews that actually fall into a particular category. It shows the proportion of reviews that actually fall into that category and are successfully detected by the model.

- Recall for Negative Sentiment

  Of all the reviews that are actually negative, the model successfully detected 87%. The calculation is:
  Recall (Negative)=3823/(3823+0+560)≈0.87
  This means that 87% of all the negative reviews were actually recognized by the model.
- Recall for Positive Sentiment

  Of all the reviews that are actually positive, the model successfully detected 94%. The calculation is:
  Recall (Positive)=13851/(13851+859+0)≈0.94
  This means that 94% of all positive reviews were actually recognized by the model.

- F1-Score

F1-Score is a combined measure of precision and recall. It gives an overall picture of how well the model classifies reviews, considering the balance between precision and recall.

- F1-Score for Negative Sentiment

  The F1-Score for negative sentiment calculates the balance between precision and recall and is 0.80. The calculation is:
  F1-Score (Negative)=2×((0.74x0.87)/(0.74+0.87))≈0.80
  This shows a good balance between precision and recall for negative reviews.

- F1-Score for Positive Sentiment
  The F1-Score for positive sentiment calculates the balance between precision and recall and the result is 0.94. The calculation is:
  F1-Score (Positive)= $2\times((0.93\times0.94)/(0.93+0.94))\approx0.94$
  This shows a very good balance between precision and recall for positive reviews.

➢ Support

Support refers to the total number of reviews in each sentiment category, providing insight into the category sizes. There are 4,383 reviews for negative sentiment, including both truly negative and misclassified reviews. For positive sentiment, there are 14,710 reviews, covering both genuinely positive and incorrectly labeled ones.

➢ Accuracy

Accuracy measures how often the model makes correct predictions overall, compared to the total number of predictions made.
The model managed to correctly classify reviews 88% of the time. The calculation is:
Accuracy = $(3823+0+13851)/(3823+0+560+475+0+431+859+0+13851) \approx 0.88$
This means that 88% of all reviews were correctly predicted by the model.
The results of the Naïve Bayes algorithm testing can be seen in table 3.2 below:

Table 2. Naïve Bayes Algorithm Testing

Algoritma	**Metrik**						**Accuracy**
	Precision			**Recall**			
	Negatif	**Netral**	**Positif**	**Negatif**	**Netral**	**Positif**	
Naïve Bayes	0.74	0	0.93	0.87	0	0.94	0.88
	F1-Score			**Support**			
	Negatif	**Netral**	**Positif**	**Negatif**	**Netral**	**Positif**	
	0.8	0	0.94	4383	907	14710	

Table 2 contains the following information:

- Precision measures the accuracy of the model in classifying reviews into the specified sentiment category. With a precision of 0.74 for negative sentiment and 0.93 for positive sentiment, the model performs well in identifying positive reviews, but is slightly less accurate in identifying negative reviews.
- Recall shows the model's ability to capture reviews that truly match the sentiment category. The model has a very high recall for positive sentiment (0.94), which means the model is very effective in detecting positive reviews. In contrast, the recall for negative sentiment is 0.87, indicating that there are some negative reviews that are not detected well by the model.
- F1-score is the harmonic average of precision and recall, providing a more comprehensive picture of the model's performance. The model has a high F1-score for positive sentiment (0.94), indicating a good balance between precision and recall for this category. The F1-score for negative sentiment is 0.80, indicating lower performance compared to positive sentiment, but still quite good.

- Support shows the number of reviews in each sentiment category. There are more positive reviews (14710) than negative reviews (4383). This reflects an imbalanced data distribution, which may affect the model's performance in detecting less frequent sentiments.
- The model accuracy is 0.88, indicating that the Naïve Bayes model is similarly effective in classifying the sentiments of reviews. This figure reflects the rate at which the model makes correct predictions out of the total predictions.

2. Decision Trees

Evaluation Metrics

Evaluation metrics are obtained from calculations using the confusion matrix value from table 1, the calculation is as follows:

➢ Precision

Precision measures how accurate the model is in classifying reviews into the correct category. This shows the proportion of reviews that truly match the category predicted by the model.

- Precision for Negative Sentiment
  Of the total reviews predicted as negative, 71% are truly negative. The calculation is:
  Precision (Negative)=3823 (True-negative reviews))/(3220+407+902)≈0.71
  This means that 71% of all reviews predicted as negative are indeed negative.
- Precision for Positive Sentiment
  Of the total reviews predicted as positive, 91% are truly positive. The calculation is:
  Precision (Positive)=13667 (True-positive reviews)/(13667+951+451)≈0.91
  This means that 91% of all reviews predicted to be positive are indeed positive.

➢ Recall

Recall measures how well the model finds all reviews that actually fall into a particular category. It shows the proportion of reviews that actually fall into that category and are successfully detected by the model.

- Recall for Negative Sentiment
  Of all the reviews that are actually negative, the model successfully detected 73%. The calculation is:
  Recall (Negative)=3220/(3220+212+951)≈0.73
  This means that 73% of all the negative reviews were actually recognized by the model.
- Recall for Positive Sentiment
  Of all the reviews that are actually positive, the model successfully detected 93%. The calculation is:
  Recall (Positive)=13667/(13667+902+141)≈0.93
  This means that 93% of all positive reviews were actually recognized by the model.

➢ F1-Score

F1-Score is a combined measure of precision and recall. It gives an overall picture of how well the model classifies reviews, considering the balance between precision and recall.

- F1-Score for Negative Sentiment
  The F1-Score for negative sentiment calculates the balance between precision and recall and is 0.72. The calculation is:
  F1-Score (Negative)=2×((0.71x0.73)/(0.71+0.73))≈0.72
  This shows a good balance between precision and recall for negative reviews.
- F1-Score for Positive Sentiment
  The F1-Score for positive sentiment calculates the balance between precision and recall and the result is 0.92. The calculation is:
  F1-Score (Positive)=2×((0.91×0.93)/(0.91+0.93))≈0.92
  This shows a very good balance between precision and recall for positive reviews.

➢ Support

Support refers to the total number of reviews in each sentiment category, providing insight into the category sizes. There are 4,383 reviews for negative sentiment, including both truly negative and misclassified reviews. For positive sentiment, there are 14,710 reviews, covering both genuinely positive and incorrectly labeled ones.

➢ Accuracy

Accuracy measures how often the model makes correct predictions overall, compared to the total number of predictions made.
The model managed to correctly classify reviews 85% of the time. The calculation is:
Accuracy = (3220+49+13667)/(3220+212+951+407+49+451+902+141+13667)≈0.85
This means that 85% of all reviews were correctly predicted by the model.
The results of the Decision Trees algorithm testing can be seen in table 3.3 below:

Table 3. Decision Trees Algorithm Testing

Algoritma	**Metrik**						**Accuracy**
	Precision			**Recall**			
	Negatif	**Netral**	**Positif**	**Negatif**	**Netral**	**Positif**	
Decision Trees	0.71	0.12	0.91	0.73	0.05	0.93	0.85
	F1-Score			**Support**			
	Negatif	**Netral**	**Positif**	**Negatif**	**Netral**	**Positif**	
	0.72	0.07	0.92	4383	907	14710	

Table 3 contains the following information:

- Precision measures the accuracy of the model in classifying reviews into the specified sentiment category. With a precision of 0.74 for negative sentiment and 0.93 for positive sentiment, the model performs well in identifying positive reviews, but is slightly less accurate in identifying negative reviews.
- Recall shows the model's ability to capture reviews that truly match the sentiment category. The model has a very high recall for positive sentiment (0.94), which means the model is very effective in detecting positive reviews. In contrast, the recall for negative sentiment is 0.87, indicating that there are some negative reviews that are not detected well by the model.

- F1-score is the harmonic average of precision and recall, providing a more comprehensive picture of the model's performance. The model has a high F1-score for positive sentiment (0.92), indicating a good balance between precision and recall for this category. The F1-score for negative sentiment is 0.72, indicating lower performance compared to positive sentiment, but still quite good.
- Support shows the number of reviews in each sentiment category. There are more positive reviews (14710) than negative reviews (4383). This reflects an imbalanced data distribution, which may affect the model's performance in detecting less frequent sentiments.
- The model accuracy is 0.85, indicating that the Decision Trees model is is also quite effective at sentiment classification. This accuracy demonstrates the proportion of correct predictions relative to the total number of predictions made by the model.

3. Support Vector Machine (SVM)

Evaluation Metrics

Evaluation metrics are obtained from calculations using the confusion matrix value from table 1, the calculation is as follows:

➢ Precision

Precision measures how accurate the model is in classifying reviews into the correct category. This shows the proportion of reviews that truly match the category predicted by the model.

- Precision for Negative Sentiment
  Of the total reviews predicted as negative, 75% are truly negative. The calculation is:
  Precision (Negative)=3788 (True-negative reviews)/(3788+466+828)≈0.75
  This means that 75% of all reviews predicted as negative are indeed negative.
- Precision for Positive Sentiment
  Of the total reviews predicted as positive, 93% are truly positive. The calculation is:
  Precision (Positive)=13881 (True-positive reviews)/(13881+594+441)≈0.93
  This means that 93% of all reviews predicted to be positive are indeed positive.

➢ Recall

Recall measures how well the model finds all reviews that actually fall into a particular category. It shows the proportion of reviews that actually fall into that category and are successfully detected by the model.

- Recall for Negative Sentiment
  Of all the reviews that are actually negative, the model successfully detected 86%. The calculation is:
  Recall (Negative)=3788/(3788+1+594)≈0.86
  This means that 86% of all the negative reviews were actually recognized by the model.
- Recall for Positive Sentiment
  Of all the reviews that are actually positive, the model successfully detected 94%. The calculation is:

Recall (Positive)=13881/(13881+828+1)≈0.94

This means that 94% of all positive reviews were actually recognized by the model.

- F1-Score

F1-Score is a combined measure of precision and recall. It gives an overall picture of how well the model classifies reviews, considering the balance between precision and recall.

- F1-Score for Negative Sentiment

  The F1-Score for negative sentiment calculates the balance between precision and recall and is 0.80. The calculation is:

  F1-Score (Negative)=2×((0.75x0.86)/(0.75+0.86))≈0.80

  This shows a good balance between precision and recall for negative reviews.
- F1-Score for Positive Sentiment

  The F1-Score for positive sentiment calculates the balance between precision and recall and the result is 0.94. The calculation is:

  F1-Score (Positive)=2×((0.93×0.94)/(0.93+0.94))≈0.94

  This shows a very good balance between precision and recall for positive reviews.

- Support

Support refers to the total number of reviews in each sentiment category, providing insight into the category sizes. There are 4,383 reviews for negative sentiment, including both truly negative and misclassified reviews. For positive sentiment, there are 14,710 reviews, covering both genuinely positive and incorrectly labeled ones.

- Accuracy

Accuracy measures how often the model makes correct predictions overall, compared to the total number of predictions made.

The model managed to correctly classify reviews 88% of the time. The calculation is:

Accuracy = (3788+0+13881)/(3788+1+594+466+0+441+828+1+13881) ≈ 0.88

This means that 88% of all reviews were correctly predicted by the model.

The results of the SVM algorithm testing can be seen in table 3.4 below:

Table 4. SVM Algorithm Testing

Algoritma	**Metrik**						**Accuracy**
	Precision			**Recall**			
	Negatif	**Netral**	**Positif**	**Negatif**	**Netral**	**Positif**	
SVM	0.75	0	0.93	0.86	0	0.94	0.88
	F1-Score			**Support**			
	Negatif	**Netral**	**Positif**	**Negatif**	**Netral**	**Positif**	
	0.8	0	0.94	4383	907	14710	

Table 4 contains the following information:

- Precision measures the accuracy of the model in classifying reviews into the specified sentiment category. With a precision of 0.75 for negative sentiment and 0.93 for positive sentiment, the model performs well in identifying positive reviews, but is slightly less accurate in identifying negative reviews.

- Recall shows the model's ability to capture reviews that truly match the sentiment category. The model has a very high recall for positive sentiment (0.94), which means the model is very effective in detecting positive reviews. In contrast, the recall for negative sentiment is 0.86, indicating that there are some negative reviews that are not detected well by the model.
- F1-score is the harmonic average of precision and recall, providing a more comprehensive picture of the model's performance. The model has a high F1-score for positive sentiment (0.94), indicating a good balance between precision and recall for this category. The F1-score for negative sentiment is 0.80, indicating lower performance compared to positive sentiment, but still quite good.
- Support shows the number of reviews in each sentiment category. There are more positive reviews (14710) than negative reviews (4383). This reflects an imbalanced data distribution, which may affect the model's performance in detecting less frequent sentiments.
- The model accuracy is 0.88, indicating that the SVM model is highly effective at determining the sentiment of reviews. This accuracy score represents the percentage of correct predictions compared to all the predictions made.

4. CONCLUSION

The study on the comparison of Naive Bayes, Decision Trees, and Support Vector Machine (SVM) algorithms for sentiment classification of JMO applications using scraping data from Google Play Store and Python applications shows interesting results. SVM shows the best performance with the highest precision, recall, and f1-score in the positive sentiment category compared to other algorithms, which are 0.93, 0.94, and 0.94, respectively. Although the precision and recall for negative sentiment are slightly lower compared to Naive Bayes, SVM still excels overall with an accuracy of 0.88. Naive Bayes shows comparable results to SVM in terms of precision, recall, and f1-score for both sentiment categories, with the same accuracy of 0.88. On the other hand, Decision Trees has a lower performance, with precision, recall, and f1-score for negative sentiment being lower than the other two algorithms, and an overall accuracy of 0.85. In conclusion, SVM and Naive Bayes are superior algorithms for the sentiment classification task of JMO applications, with SVM showing a slight advantage in terms of f1-score and accuracy. Decision Trees, although simpler, show less optimal results in this sentiment classification context.

5. SUGGESTED

Based on the results of the study comparing the Naive Bayes, Decision Trees, and SVM algorithms for sentiment classification of JMO applications, it is recommended to use the SVM algorithm in practical applications. SVM showed superior performance with the highest precision, recall, and f1-score on positive sentiment, as well as better overall accuracy compared to Naive Bayes and Decision Trees. Although Naive Bayes also showed solid results and was on par with SVM in several metrics, SVM provided additional advantages in terms of accuracy and f1-score, especially in the positive sentiment category. Decision Trees, although often used due to its simplicity and interpretability, proved to be less than optimal in

terms of performance compared to the other two algorithms, especially in precision and recall for negative sentiment. Therefore, for applications that require high accuracy and better classification capabilities, SVM is a better choice. However, if model interpretability is a primary consideration, Naive Bayes can be a good alternative with adequate performance. The final decision should consider the balance between accuracy and the need for model interpretability, as well as the characteristics of the available data.

6. REFERENCES

[1] R. Maulana, A. Voutama, and T. Ridwan, "Analisis Sentimen Ulasan Aplikasi MyPertamina pada Google Play Store menggunakan Algoritma NBC," *J. Teknol. Terpadu*, vol. 9, no. 1, pp. 42–48, 2023, doi: 10.54914/jtt.v9i1.609.

[2] F. A. Larasati, D. E. Ratnawati, and B. T. Hanggara, "Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest," *... Teknol. Inf. dan ...*, vol. 6, no. 9, pp. 4305–4313, 2022, [Online]. Available: http://j-ptiik.ub.ac.id

[3] A. Muhammadin and I. A. Sobari, "Analisis Sentimen Pada Ulasan Aplikasi Kredivo Dengan Algoritma Svm Dan Nbc," *Reputasi J. Rekayasa Perangkat Lunak*, vol. 2, no. 2, pp. 85–91, 2021, doi: 10.31294/reputasi.v2i2.785.

[4] M. K. Insan, U. Hayati, and O. Nurdiawan, "Analisis Sentimen Aplikasi Brimo Pada Ulasan Pengguna Di," *J. Mhs. Tek. Inform.*, vol. 7, no. 1, pp. 478–483, 2023.

[5] H. Z. Muflih, A. R. Abdillah, and F. N. Hasan, "Analisis Sentimen Ulasan Pengguna Aplikasi Ajaib Menggunakan Metode Naïve Bayes," *KLIK Kaji. Ilm. Inform. dan Komput.*, vol. 4, no. 3, pp. 1613–1621, 2023, doi: 10.30865/klik.v4i3.1303.

[6] Friska Aditia Indriyani, Ahmad Fauzi, and Sutan Faisal, "Analisis sentimen aplikasi tiktok menggunakan algoritma naïve bayes dan support vector machine," *TEKNOSAINS J. Sains, Teknol. dan Inform.*, vol. 10, no. 2, pp. 176–184, 2023, doi: 10.37373/tekno.v10i2.419.

[7] Ernianti Hasibuan and Elmo Allistair Heriyanto, "Analisis Sentimen Pada Ulasan Aplikasi Amazon Shopping Di Google Play Store Menggunakan Naive Bayes Classifier," *J. Tek. dan Sci.*, vol. 1, no. 3, pp. 13–24, 2022, doi: 10.56127/jts.v1i3.434.