Jurnal Varian
Vol. 7, No. 2, April 2024, Hal. 115–123
e-ISSN: 2581-2017

Comparison of Naive Bayes Classification Methods Without and With Kernel Density Estimation

Agus Hermawan[1], **Siswanto Siswanto**[1], **Andi Kresna Jaya**[1]
[1]Department of Statistics, Faculty of Mathematics and Natural Sciences, Universitas Hasanuddin, Indonesia

Article Info

Article history:

Received : xx
Revised : xx
Accepted : 10-19-2023

Keywords:

Classification;
Halal Certification;
Kernel Density Estimation;
Naive Bayes Algorithm;
Self Declare.

ABSTRACT

This study aims to address the need for verification and validation of data for halal certification applications in Indonesia by using the data science approach and machine learning technology. The method used in this study is the Naive Bayes classification to optimize the data verification and validation process. However, this method needs to be enhanced by applying optimization techniques such as Kernel Density Estimation (KDE) to improve the classification results. The results showed that the Naive Bayes classification method with KDE optimization produced better performance than the Naive Bayes method without optimization. The performance of the Naive Bayes classification model without optimization achieves 87.6% Accuracy, 85.4% Recall, 88.8% Precision, and 87.1% Fmeasure. Meanwhile, the Naive Bayes classification model with KDE optimization achieves 97.5% Accuracy, 95.9% Recall, 98.9% Precision, and 97.8% F-measure. Thus, it can be concluded that the Naive Bayes classification algorithm with KDE optimization results in a performance increase of 9.9% compared to the Naive Bayes method without optimization. This research has important implications in handling complex and non-normally distributed data and providing solutions for BPJPH in the process of verifying halal certification. Additionally, this study contributes to the broader discourse on the integration of advanced computational methods in enhancing certification processes. By bridging cutting-edge computational techniques with traditional certification practices, our research not only improves efficiency but also fosters greater transparency and trust in the halal market. The successful application of KDE optimization to the Naive Bayes classification method offers a promising pathway towards a more efficient and technologically-driven certification framework within the Halal Product Assurance Organizing Agency (BPJPH) in Indonesia. To build on this success, it is recommended to explore alternative classification methods such as Support Vector Machines (SVM), Decision Trees, or Neural Networks, allowing for a comprehensive comparison of performance and effectiveness against the optimized Naive Bayes method. Additionally, applying cross-validation techniques like k-fold cross-validation can help mitigate overfitting and provide a more accurate estimate of model performance on unseen data. Further research into optimal parameter settings for both Naive Bayes and KDE could also enhance model performance significantly.

Accredited by Kemenristekdikti, Decree No: 200/M/KPT/2020
DOI: https://doi.org/10.30812/varian.v7i2.3199

Corresponding Author:

Siswanto Siswanto,
Department of Statistics, Universitas Hasanuddin
Email: siswanto@unhas.ac.id

A. INTRODUCTION

Ensuring the authenticity of halal products is paramount in instilling trust among Muslim consumers worldwide. Indonesia, with its status as the country with the largest Muslim population, stands at the forefront of this endeavour. The Halal Product Assurance Organizing Agency (BPJPH), established by the Indonesian government, plays a pivotal role in certifying products' halal status. However, the manual verification and validation processes employed by BPJPH have become increasingly inefficient in coping with the rising number of certification applications (Mohammad, 2021). As the demand for halal products continues to surge, there arises

an urgent need for innovative solutions to streamline and enhance the certification procedures.

In recent years, advancements in data science and machine learning have presented promising opportunities for optimizing various processes across industries. Recognizing this potential, our study seeks to harness these technological advancements to revolutionize the halal certification landscape in Indonesia. By employing sophisticated computational methods, particularly the Naive Bayes classification algorithm and Kernel Density Estimation (KDE), we aim to enhance the accuracy and efficiency of halal certification verification and validation processes conducted by BPJPH.

Building upon previous research endeavours that have explored the application of machine learning techniques in diverse domains (Nugroho et al., 2019) and The Sentiment Analysis Using Naive Bayes with Lexicon-Based Feature on TikTok Application (Siswanto et al., 2022), our study focuses on adapting these methodologies to address the unique challenges within the halal certification framework. Specifically, we aim to evaluate the effectiveness of integrating KDE as an optimization technique for the Naive Bayes classification method in processing halal certification applications (Bullmann et al., 2018). By doing so, we aspire to provide BPJPH with a robust and scalable solution capable of handling the complexities of certification data while ensuring swift and accurate decision-making. However, the Naive Bayes method has drawbacks when dealing with datasets with non-uniform attribute weights, as well as strong assumptions about independence between features (Kashif et al., 2021).

The novelty of our approach lies in its intersection of cutting-edge technology with the traditional domain of halal certification. By bridging these realms, we not only seek to enhance the efficiency of certification processes but also contribute to bolstering consumer confidence in halal products (Bakar and Rosbi, 2019). Through this research endeavor, we endeavor to pave the way for a more streamlined and technologically-driven approach to halal certification in Indonesia, thereby fostering greater transparency and trust in the halal market (Tarannum, 2023).

To overcome this, this study used Kernel Density Estimation (KDE) as an optimization method for the Naive Bayes algorithm. KDE is a statistical method for estimating the probability density function of a group of data by finding the best kernel . The use of KDE in text data with a large amount of data is very suitable to overcome the problem of halal certification verification at BPJPH. Thus, this study aims to compare the performance of the Naive Bayes classification method without and with KDE in the 2022 BPJPH Self Declare data, in the hope of finding a more optimal solution in overcoming complex and non-normally distributed data.

B. RESEARCH METHOD

1. Indonesian Halal Certification

In Indonesia, halal certification requirements are limited to various ingredients used in the production of halal products. These materials include animal, plant, microbial, and materials produced through chemical, biological, or genetic engineering processes. Every material used in the manufacture of halal products, whether as raw materials, processed materials, additives, or auxiliary materials, must meet halal criteria in accordance with religious principles. Self-declare is an Unpaid Halal Certification program provided by the government to accelerate the achievement of the target of 10 million halal-certified products. The criteria for products that fall into the self-declare category are contained in the Decree of the Head of BPJPH Number 33 of 2022 concerning Self-declare Criteria.

2. Naive Bayes

The Naive Bayes algorithm, which is based on Bayes' Theorem, is used in text mining to classify documents into specific classes (Delizo et al., 2020). Naive Bayes' assumption that each feature (word) in the document is independent of each other provides a faster and simpler training process that requires less training data (Sen et al., 2020). However, this assumption can also cause classification results to be less accurate if the features in the document are interrelated (Ali et al., 2020). The conditional chance of an event occurring provided that another event has already occurred can be calculated using a predetermined formula. Here is the conditional probability formula later used in Bayes Theorem Equation (1) (Delizo et al., 2020):

$$P(C|X) = \frac{P(X \cap C)}{P(X)}, \quad P(X) > 0 \tag{1}$$

Information:

$P(C\|X)$	:	Opportunity of occurrence C on condition that it X has happened
$P(X)$	:	Opportunity of occurrence X
$P(X \cap C)$	:	Opportunity of occurrence X on condition that it C has happened

To determine the probability of features x_i in the C class, it is written in Equation (2) as follows:

$$\begin{aligned} P(C|X_1) &= \frac{P(X_1 \cap C)}{P(X_1)} \\ P(C|X_2) &= \frac{P(X_2 \cap C)}{P(X_2)} \\ &\vdots \\ P(C|X_n) &= \frac{P(X_n \cap C)}{P(X_n)} \end{aligned} \tag{2}$$

From the description above, it can be seen that the more and more complex the condition factors that affect the value of opportunities, making it difficult to analyze one by one. Therefore, the assumption of very high independence (naive) is used to facilitate the calculation, that each is independent of each $(X_1, X_2, X_3, \ldots, X_n)$ other. Thus, the following Equation (3) apply:

$$\begin{aligned} P(X_a|X_b) &= \frac{P(X_a \cap X_b)}{P(X_b)} \\ &= \frac{P(X_a) \times P(X_b)}{P(X_b)} \\ &= P(X_a) \end{aligned} \tag{3}$$

So, based on this equation, the probability requirements become simpler and the calculation become possible. Furthermore, $P(C|X)$ using the multiplication rules can be decomposed into

$$\begin{aligned} P(C|X) &\propto P(X \cap C) \\ &= P(X_1 \cap X_2 \cap X_3 \cap \ldots \cap X_n \cap C) \\ &= P(X_1|X_2 \cap X_3 \cap \ldots \cap X_n \cap C) \times P(X_2 \cap X_3 \cap \ldots \cap X_n \cap C) \\ &= Pp(X_1|X_2 \cap X_3 \cap \ldots \cap X_n \cap C) \times P(X_2|X_3 \cap \ldots \cap X_n \cap C) \times P(X_3 \cap \ldots \cap X_n \cap C) \end{aligned}$$

To determine the opportunity of features x_i in the C class are as follows Equation (4):

$$\begin{aligned} P(C|X_1, X_2, X_3, \ldots X_n) &\propto P(C)P(X_1|C)P(X_2|C) \cdots P(X_n|C) \\ &= P(C) \times \prod_{i=1}^{n} P(X_i|C) \end{aligned} \tag{4}$$

This equation is also referred to as the posterior equation which is obtained from the product between the prior and likelihood values.

Kernel Density Estimation (KDE) is a non-parametrial statistical method used to derive estimates from density functions and estimate probability distributions from data (Haben and Giasemidis, 2016). KDE can be used as an optimization of Naive Bayes, which is a classification algorithm popularly used in text mining (Ji et al., 2019). Naive Bayes considers that all words in a document are independent of each other, while KDE takes the correlation between words in the document into calculation. By using KDE, the Naive Bayes model is strengthened and can provide more accurate results (Qin and Xiao, 2018).

KDE optimizes the Naive Bayes algorithm by providing a better estimation of the probability of words in a document by taking the correlation between words in the document into calculations (Weglarczyk, 2018). Here is the general equation for calculating KDE weights in Equation (5):

$$KDE_i = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{\widehat{x} - \widehat{x}_i}{h}\right) \tag{5}$$

Information:

KDE_i	:	Weights
n	:	Lots of data or features
$\widehat{x}$	:	Data point probability value
$\widehat{x}_i$	:	i-th data probability value of n

DOI: https://doi.org/10.30812/varian.v7i2.3199

Estimator of $\widehat{x}$ calculated using the following Equation (6):

$$\widehat{x} = \frac{m_i}{M} \tag{6}$$

Information:
m_i : number of occurrences of a word in the corpusi
M : the number of occurrences of all words in the corpus

Then the estimator of $\widehat{x}_i$ calculated using the following Equation (7):

$$\widehat{x}_i = \frac{m_i}{N} \tag{7}$$

Information:
N : The number of documents

For h is bandwidth using the Maximum Likelihood Estimation function in Equation (8):

$$h = 0,01\sigma n^{\frac{1}{5}} \tag{8}$$

Information:
σ : Sigma is the standard deviation of the data

for K is the kernel Gaussian function in Equation (9):

$$K_i = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(\widehat{x} - \widehat{x}_i)^2}{2h^2}\right) \tag{9}$$

3. Data

The data used in this study is secondary data derived from the submission of halal certification by business actors through the self-declare scheme provided by the Halal Product Assurance Organizing Agency (BPJPH) located in East Jakarta, DKI Jakarta in the 2022 fiscal year. This data is in the form of text consisting of a list of product names submitted by business actors to meet the requirements for halal certification. Each product name data is labelled ”Positive” or ”Negative” based on the criteria set by the head of BPJPH. Product names labelled ”Positive” mean that they have been verified and are eligible to apply for halal certification free of charge through the self-declaration scheme, while product names labelled ”Negative” do not meet the verification of halal certification submissions through the self-declaration scheme. The total amount of data to be used in this study is 20,000, and will be divided into 90% training data and 10% test data.

This 90-10 split is a standard practice in machine learning to ensure a robust training process while retaining enough data for meaningful evaluation of the model’s performance. With a dataset of 20,000 records, allocating 90% (18,000 records) for training allows the model to learn a wide range of patterns and nuances within the data, thereby improving its generalization capabilities. The remaining 10% (2,000 records) is sufficient to conduct a thorough and statistically significant evaluation of the model, as it provides ample data to assess the models performance on unseen data. This balance ensures that the model is well-trained and that its evaluation on test data gives a reliable indication of how it will perform in real-world applications.

The data was analysed using a machine learning approach technique with a text mining model, Naive Bayes algorithm, as a text data classification method with KDE feature extraction as model optimization. Figure 1 is the flowchart analysis and the detailed stages of the data processing process in this study:

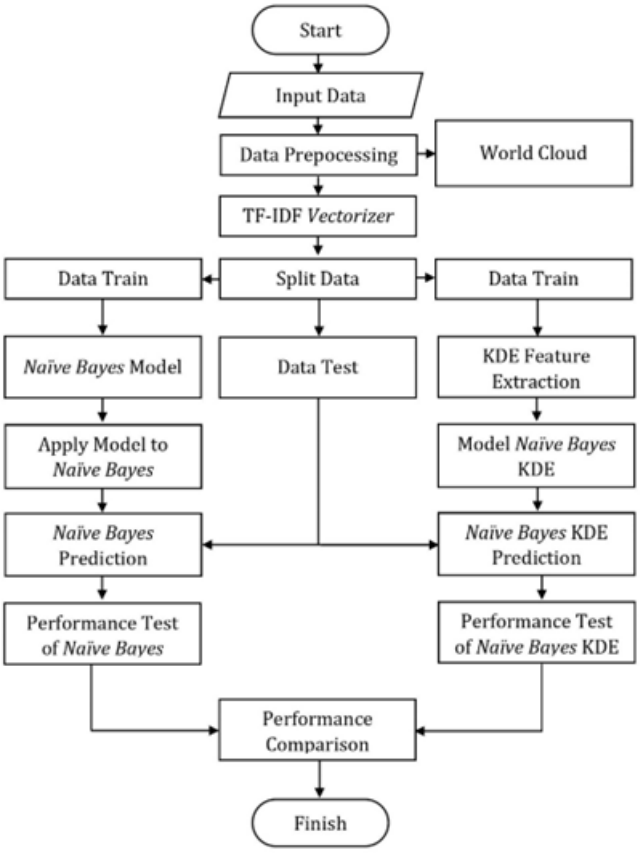

Figure 1. Research Flowchart: Comparing Naive Bayes with and without KDE Optimization

The Figure 1 shows the flow of research implementation in the data processing section with the aim of comparing two classification methods. The following is a more detailed explanation of the flow chart above:

1. Data collection of halal certification submission for business actors through a self-declare scheme from BPJPH's database of 20,000 data.
2. Perform data pre-processing (data preprocessing). In this study, the data preprocessing process includes the following steps: punctuation removal, case folding, tokenizing, and stopwords removal
3. Convert text to vector numeric features with TF-IDF Vectorizer method.
4. Division of data into training data and test data using Train/Test split.
5. Train the data to build a model with the Naive Bayes algorithm on the training data to then use the model to perform predictions.
6. Test the percentage of model performance based on Accuracy, Precision, Recall, and Fmesure value. These metrics are chosen to provide a balanced assessment of the model's accuracy, its ability to correctly identify positives, and its overall capability to capture relevant instances while considering both false positives and false negatives.
7. Feature extraction and normalization are performed by the KDE using the following Equation (10) (Silverman, 2018):

$$KDE_i = \frac{1}{nh\sqrt{2\pi}} \sum_{i=1}^{n} \exp\left(-\frac{(\widehat{x} - \widehat{x}_i)^2}{2h^2}\right) \tag{10}$$

and normalization is carried out using the following Equation (11):

$$P(KDE_i) = \frac{KDE_i + k}{\sum_i^n KDE_i + kn} \tag{11}$$

8. Apply the extracted feature to the Naive Bayes prediction model and perform predictions using the following Equation (12):

$$P(C|X_1, \ldots, X_n) = P(C) \times \prod_{i=1}^{n} P(X_i|C) \tag{12}$$

DOI: https://doi.org/10.30812/varian.v7i2.3199

9. Test the percentage of model performance based on Accuracy, Precision, Recall, and $F_{measure}$.
10. Display test data results with label prediction results

C. RESULT AND DISCUSSION

1. Data Description

The data in this study was obtained from the submission of halal certification owned by BPJPH in 2022 through self-declare. The data consists of 20,000 "Product" and "Label" columns on Table 1.

Table 1. Dataset

No	Product	Label
1	Makanan Kering Yang Meliputi Keripik	Positive
2	Peternakan Ayam @ Petelor	Negative
3	Honny & Lemon	Positive
4	Juice Buah Buahan%!	Positive
5	Rose Catering di Jln Bandes Taratak	Positive
⋮	⋮	⋮
19998	Obat Obat Herbal	Negative
19999	Sambal Bajak	Positive
20000	Sapi Potong	Negative

This data consists of 9,989 or 49.94% data labeled Positive which means product submissions can be made through free halal certification and 10,011 or 50.06% data labeled Negative which means product submissions must go through the paid registration route.

2. Punctuation Removal

Stages of the process of removing punctuation from a text or sentence. Punctuation marks such as periods, question marks, and others. The results of the punctuation removal stage are in Table 2:

Table 2. Punctuation removal result

Before	After
Makanan Kering Yang Meliputi Keripik	Makanan Kering Yang Meliputi Keripik
Peternakan Ayam @ Petelor	Peternakan Ayam Petelor
Honny & Lemon	Honny Lemon
Juice Buah Buahan%!	Juice Buah Buahan
⋮	⋮
Sapi Potong	Sapi Potong

3. Case Folding

The process stages of converting characters in text to lowercase or capital letters for the purpose of consistency and ease of processing and eliminating the distinction between upper and lower case letters in the text. The results of the case folding stages are in Table 3:

Table 3. Case folding result

Before	After
Makanan Kering Yang Meliputi Keripik	makanan kering yang meliputi keripik
Peternakan Ayam Petelor	peternakan ayam petelor
Honny Lemon	honny lemon
Juice Buah Buahan	juice buah buahan
⋮	⋮
Sapi Potong	sapi potong

DOI: https://doi.org/10.30812/varian.v7i2.3199

4. Stopwords Removal

The stages of the process eliminate connecting or general words that do not give significant meaning to the text. These words usually consist of words such as ”which”, ”and”, ”for”, and others. The results of the stopwords removal stages are in Table 4:

Table 4. Stopwords removal result

Before	After
makanan kering yang meliputi keripik	makanan kering meliputi keripik
peternakan ayam petelor	peternakan ayam petelor
honny lemon	honny lemon
juice buah buahan	juice buah buahan
⋮	⋮
sapi potong	sapi potong

5. Tokenizing

The process of breaking down text or sentences into smaller parts is called tokenize or tokenize. The results of the stopwords removal stages are in Table 5:

Table 5. Tokenizing result

Before	After
makanan kering meliputi keripik	[makanan, kering, meliputi, keripik]
peternakan ayam petelor	[peternakan, ayam, petelor]
honny lemon	[honny, lemon]
juice buah buahan	[juice, buah, buahan]
⋮	⋮
sapi potong	[sapi, potong]

6. Word Cloud

A technique used to visualize a number of texts by displaying the words that most often appear in the form of visual nuances. The results of the Word Cloud visualization are shown in Figure 2:

Figure 2. Word Cloud

7. Count Vectorizer

The stage counts the number of occurrences of each word in a document or document set and produces a matrix consisting of documents as rows and words as columns. Each cell in the matrix indicates the number of occurrences of the word in the corresponding document. Table 6 is the result of the vectorizer count of the occurrence of each feature in the entire dataset:

Table 6. Tokenizing result

No	Product	Ayam	⋯	Narsi
1	makanan kering ...	0	⋯	0
2	peternakan ayam ...	1	⋯	0
3	honny lemon	0	⋯	0
4	juice buah buahan	0	⋯	0
5	rose catering ...	0	⋯	0
⋮	⋮	⋮	⋮	⋮
19999	sambal bajak	0	⋯	0
20000	sapi potong	0	⋯	0

The dataset is divided by a proportion of 90% for the training data and 10% for the test data. Table 7 is a table of proportions of division of training data and test data:

Table 7. Proportion of data sharing

Data	Class		Total
	Positif	Negatif	
Data train	8.967	9.033	18
Data test	1.022	978	2
Total	9.989	10.011	20

TF-IDF Vectorizer on Train Data. TF-IDF Vectorizer assigns weights based on the number of words they appear in the document and how many documents contain those words. Table 8 is the result of the vectorizer count of the occurrence of each feature in the training data:

Table 8. Count vectorizer data train

No	Product	Ayam	⋯	Lable
1	makanan kering	0	⋯	Positive
2	peternakan ayam	1	⋯	Negative
3	honny lemon	0	⋯	Positive
4	juice buah buahan	0	⋯	Positive
5	rose catering ...	0	⋯	Positive
⋮	⋮	⋮	⋮	⋮
18000	jal mie ayam	1	⋯	Negative

Then, features are taken with a difference of occurrence of at least 30 in both classes on the grounds that features that have high difference values contribute well to classification compared to features that only have low difference values. Table 9 is the difference in occurrence of the training data feature:

Table 9. Feature with a difference of at least 30

No	Term	Positive	Negative	Difference
1	ayam	0	1.668	1.668
2	kue	990	75	915
3	nasi	272	887	615
4	jual	0	504	504
5	bahan	14	445	431
⋮	⋮	⋮	⋮	⋮
174	empek	38	8	30
175	bakwan	32	2	30
176	mi	19	49	30
Total		10.999	17.134	20.387

DOI: https://doi.org/10.30812/varian.v7i2.3199

The results of TF-IDF on Positive Class are shown in Table 10:

Table 10. Positive TF-IDF Results

No	Term	TF-IDF	P(TF-IDF)
1	ayam	18,4165	$6,3829 \times 10^{-3}$
2	kue	19,6747	$6,7965 \times 10^{-3}$
3	nasi	20,3764	$7,0272 \times 10^{-3}$
4	jual	21,2863	$7,3263 \times 10^{-3}$
5	bahan	21,3888	$7,3600 \times 10^{-3}$
⋮	⋮	⋮	⋮
174	empek	12,1407	$4,3198 \times 10^{-3}$
175	bakwan	6,6932	$2,5291 \times 10^{-3}$
176	mi	18,5638	$6,4313 \times 10^{-3}$

The results of TF-IDF on Negative Class are shown in Table 11:

Table 11. Negative Class TF-IDF Results

No	Term	TF-IDF	P(TF-IDF)
1	ayam	18,4165	$6,3829 \times 10^{-3}$
2	kue	19,6747	$6,7965 \times 10^{-3}$
3	nasi	20,3764	$7,0272 \times 10^{-3}$
4	jual	21,2863	$7,3263 \times 10^{-3}$
5	bahan	21,3888	$7,3600 \times 10^{-3}$
⋮	⋮	⋮	⋮
174	empek	12,1407	$4,3198 \times 10^{-3}$
175	bakwan	6,6932	$2,5291 \times 10^{-3}$
176	mi	18,5638	$6,4313 \times 10^{-3}$

8. Naive Bayes Classification

Classification with Naive Bayes on each prediction to a particular class. The results of classification with Naive Bayes in Table 12:

Table 12. Naive Bayes prediction results

No	Product	Prediction	Actual
1	p n d	Positive	Negative
2	martabak telor	Negative	Negative
3	fruit salad	Positive	Positive
4	kue basah nasi kotak tumpeng	Positive	Positive
5	pembuatan sandwich	Positive	Negative
⋮	⋮	⋮	⋮
1998	obat obat herb	Negative	Negative
1999	sambal bajak	Positive	Positive
2000	sapi potong	Negative	Negative

9. Naive Bayes Classification Performance Test

The following are the results of the Naive Bayes classification performance test based on the results of the Confusion Matrix in Figure 3 obtained:

DOI: https://doi.org/10.30812/varian.v7i2.3199

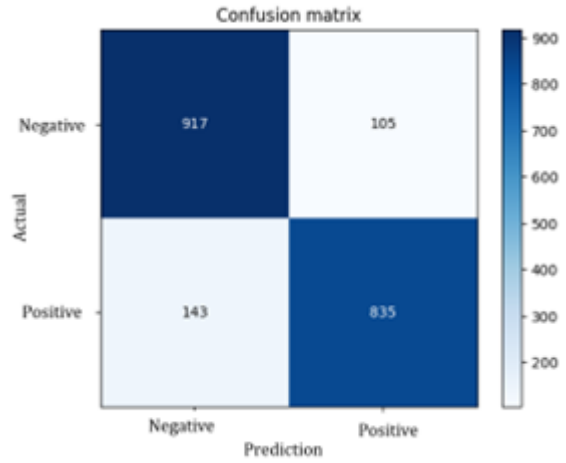

Figure 3. Naive Bayes Confusion Matrix

Thus, the performance value of Accuracy 87.6%, Precision 87.6%, Recall 85.38%, and Fmeasure 88.83%. Accuracy is an important benchmark in evaluating the performance of classification models because it can indicate the ability of the model to predict correctly. This Naive Bayes classification model has an Accuracy of 87.6%, which means that there are 12.4% of test data classified incorrectly by the model.

10. Feature Extraction With KDE

Feature extraction with KDE is one of the non-parametric statistical techniques used to estimate the probability distribution of a data. The main purpose of feature extraction with KDE is to describe the numerical distribution of data, which can be used to understand patterns and relationships between variables in the data. The process of extracting features with KDE involves establishing probability distribution models for numerical data. Based on the results from Table 9, weighting is carried out for each feature contained in the training data in each classification class using the equation (10) and normalization is carried out using Equation (11) (Silverman, 2018).

Thus, in the Positive Class the results are obtained in Table 13:

Table 13. KDE Weights on Positive Class

No	Term	KDE_{Pos}	P(KDE)_{Pos}
1	Ayam	$3,8055 \times 10^{-5}$	$5,6628 \times 10^{-3}$
2	Kue	$4,0451 \times 10^{-3}$	$5,6855 \times 10^{-3}$
3	Nasi	$3,8055 \times 10^{-5}$	$5,6628 \times 10^{-3}$
4	Jual	$1,0171 \times 10^{-4}$	$5,6632 \times 10^{-3}$
5	Bahan	$1,5818 \times 10^{-4}$	$5,6635 \times 10^{-3}$
⋮	⋮	⋮	⋮
172	Manisan	$3,9187 \times 10^{-3}$	$5,6848 \times 10^{-3}$
173	Batagor	$4,4472 \times 10^{-3}$	$5,6878 \times 10^{-3}$
174	Empek	$4,0782 \times 10^{-3}$	$5,6857 \times 10^{-3}$
175	bakwan	$3,9187 \times 10^{-3}$	$5,6848 \times 10^{-3}$
176	mi	$4,4343 \times 10^{-3}$	$5,6877 \times 10^{-3}$

The KDE_Pos column is the estimated result and probability value of the KDE distribution for each feature in the Positive Class, while the P(KDE)_Pos column is the probability value calculated from the KDE_Pos. In general, the higher the P(KDE) value_Pos, the higher likelihood that the feature is derived from Positive Class. Then, the Negative Class is obtained in Table 14:

Table 14. KDE Weights on Negative Class

No	Term	KDE_{Neg}	P(KDE)_{Neg}
1	Ayam	$6,2709 \times 10^{-3}$	$5,6859 \times 10^{-3}$
2	Kue	$6,0418 \times 10^{-5}$	$5,6508 \times 10^{-3}$

DOI: https://doi.org/10.30812/varian.v7i2.3199

3	Nasi	$5,0188 \times 10^{-4}$	$5,6533 \times 10^{-3}$
4	Jual	$6,2709 \times 10^{-3}$	$5,6859 \times 10^{-3}$
5	Bahan	$7,0909 \times 10^{-4}$	$5,6905 \times 10^{-3}$
⋮	⋮	⋮	⋮
172	Manisan	$7,2276 \times 10^{-3}$	$5,6913 \times 10^{-3}$
173	Batagor	$7,1877 \times 10^{-3}$	$5,6911 \times 10^{-3}$
174	Empek	$7,0505 \times 10^{-3}$	$5,6903 \times 10^{-3}$
175	bakwan	$7,2276 \times 10^{-3}$	$5,6913 \times 10^{-3}$
176	mi	$7,2356 \times 10^{-3}$	$5,6914 \times 10^{-3}$

The KDE_Neg column is the estimated result and probability value of the KDE distribution for each feature in the Negative Class, while the P(KDE)_Neg column is the probability value calculated from the KDE_Neg. In general, the higher the P(KDE) value_Neg, the higher the likelihood that the feature is from a Negative Class.

11. Naive Bayes KDE Classification

The process of classifying with Naive Bayes KDE on each prediction to a particular class uses the same method as Naive Bayes classification, only using weights that have been extracted features with KDE. The results of classification with Naive Bayes KDE can be seen in Table 15:

Table 15. Naive Bayes KDE Results

No	Product	Prediction	Actual
1	p n d	Negative	Negative
2	martabak telor	Negative	Negative
3	fruit salad	Positive	Positive
4	kue basah nasi kotak tumpeng	Positive	Positive
5	pembuatan sandwich	Negative	Negative
⋮	⋮	⋮	⋮
1996	kripik kebab maharani	Negative	Negative
1997	Siomay	Negative	Negative
1998	obat obat herb	Negative	Negative
1999	sambal bajak	Positive	Positive
2000	sapi potong	Negative	Negative

Product prediction results have variations and are classified into Positive or Negative Class using the Naive Bayes KDE method. Information about the performance of specific classification models cannot be known, so it is necessary to further review Accuracy, Precision, Recall, and F_{mesure}.

12. Naive Bayes KDE Classification Performance Test

The following are the results of the Naive Bayes KDE classification performance test based on the results of the Confusion Matrix in Figure 4 obtained:

DOI: https://doi.org/10.30812/varian.v7i2.3199

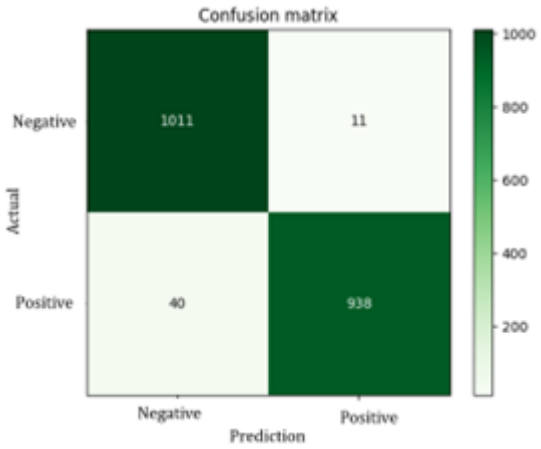

Figure 4. Naive Bayes KDE Confusion Matrix

Thus, from the results of the Naive Bayes KDE Confusion Matrix, the performance value of Accuracy 97.45%, Precision 95.91%, Recall 98.84%, and Fmeasure 97.35%. This Naive Bayes KDE classification model has an Accuracy improvement of 9.9% which means it is better than the previous one.

13. Comparison Performance Test

The following Figure 5 are the results of a comparison of performance tests on both classification methods, namely Naive Bayes and Naive Bayes KDE:

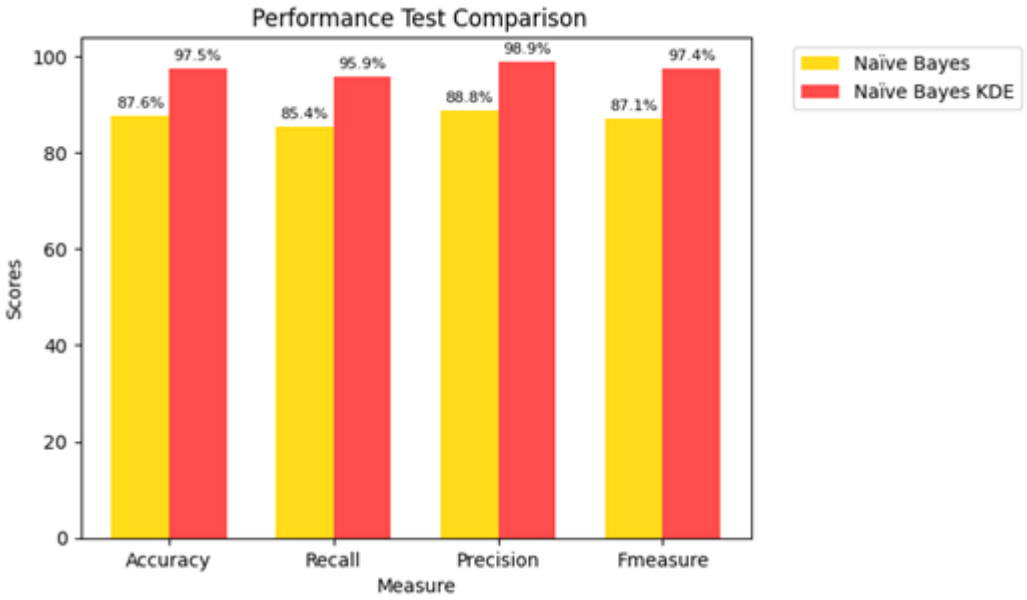

Figure 5. Performance Test Comparison

The test results using 2,000 Test Data showed that the results of Naive Bayes prediction with KDE optimization were better than the results of Naive Bayes classification testing. This can be seen from the values of Accuracy, Recall, Precision, and Fmeasure showing higher values than Naive Bayes. Thus, it can be concluded that the Naive Bayes KDE classification method gives better results in predicting Product Name data into Positive Class and Negative Class categories.

Compared to previous research that has explored the use of Naive Bayes classification methods in different contexts such as sentiment analysis on social media (Nugroho et al., 2019) and TikTok applications (Siswanto et al., 2022) our study stands out for its focus on optimizing the halal certification process in Indonesia. Specifically, this research introduces the use of Kernel Density Estimation (KDE) techniques as an optimization method to enhance the classification of halal certification application

data. While previous studies tended to focus more on data analysis and sentiment, our approach explores the application of the latest technology to directly improve the efficiency of the halal certification process.

By combining an understanding of traditional needs in halal certification with technological advancements, our research creates a convergence point that merges old practices with modern solutions. This significantly contributes to the development of the halal certification industry by offering a more integrated and technology-driven approach. Thus, our research paves the way for a more efficient and technology-driven approach to the halal certification process in Indonesia, ultimately aiming to strengthen transparency and trust in the halal market.

D. CONCLUSION AND SUGGESTION

In conclusion, our study highlights the effectiveness of the Naive Bayes KDE classification algorithm in enhancing the performance of halal certification verification processes. Building upon the traditional Naive Bayes method, the incorporation of Kernel Density Estimation (KDE) optimization significantly improves classification accuracy, as evidenced by the substantial increase in performance metrics. The Naive Bayes KDE algorithm, by applying probability density functions to each feature and considering the contribution of individual data points, achieves remarkable accuracy levels, with an Accuracy value of 97.5%, Recall of 95.9%, Precision of 98.9%, and $F_{measure}$ e of 97.8%. These results demonstrate the robustness and reliability of our approach in handling complex and non-normally distributed halal certification data.

Furthermore, our research underscores the pivotal role of technology in modernizing and streamlining halal certification processes. By bridging cutting-edge computational techniques with traditional certification practices, we not only improve efficiency but also foster greater transparency and trust in the halal market. The successful application of KDE optimization to the Naive Bayes classification method offers a promising pathway towards a more efficient and technologically-driven certification framework within the Halal Product Assurance Organizing Agency (BPJPH) in Indonesia.

In essence, our study contributes to the ongoing discourse on the integration of advanced computational methods in enhancing certification processes, with particular relevance to the halal industry. Moving forward, we envision further exploration and refinement of these methodologies to continually improve the reliability and efficiency of halal certification procedures, ultimately ensuring the integrity and authenticity of halal products in the global market.

REFERENCES

Ali, Z. M., Hassoon, N. H., Ahmed, W. S., and Abed, H. N. (2020). The application of data mining for predicting academic performance using k-means clustering and naïve bayes classification. *International Journal of Psychosocial Rehabilitation*, 24(03):2143–2151. https://doi.org/10.37200/IJPR/V24I3/PR200962.

Bakar, N. A. and Rosbi, S. (2019). Robust framework of halal certification process with integration of artificial intelligent method. *Journal of Islamic, Social, Economics and Development (JISED)*, 4(20):47–55.

Bullmann, M., Fetzer, T., Ebner, F., Deinzer, F., and Grzegorzek, M. (2018). Fast kernel density estimation using gaussian filter approximation. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1233–1240. IEEE. https://doi.org/10.23919/ICIF.2018.8455686.

Delizo, J. P. D., Abisado, M. B., and De Los Trinos, M. I. P. (2020). Philippine twitter sentiments during covid-19 pandemic using multinomial naïve-bayes. *International Journal*, 9(1.3). https://doi.org/10.15294/rji.v1i2.68324.

Haben, S. and Giasemidis, G. (2016). A hybrid model of kernel density estimation and quantile regression for gefcom2014 probabilistic load forecasting. *International Journal of Forecasting*, 32(3):1017–1022. https://doi.org/10.1016/j.ijforecast.2015.11.004.

Ji, H., Huang, S., Lv, X., Wu, Y., and Feng, Y. (2019). Empirical studies of a kernel density estimation based naive bayes method for software defect prediction. *IEICE TRANSACTIONS on Information and Systems*, 102(1):75–84. https://doi.org/10.1587/transinf.2018EDP7177.

Kashif, A. A., Bakhtawar, B., Akhtar, A., Akhtar, S., Aziz, N., and Javeid, M. S. (2021). Treatment response prediction in hepatitis c patients using machine learning techniques. *International Journal of Technology, Innovation and Management (IJTIM)*, 1(2):79–89. https://doi.org/10.54489/ijtim.v1i2.24.

Mohammad, M. F. M. (2021). The pengaturan sertifikasi jaminan produk halal di indonesia. *Kertha Wicaksana*, 15(2):149–157. https://doi.org/10.22225/kw.15.2.2021.149-157.

Nugroho, A., Hidayatillah, R., Sumpeno, S., and Purnomo, M. H. (2019). Klasifikasi interaksi kampanye di media sosial menggunakan naïve bayes kernel estimator. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, 8(2):107–114.

Qin, B. and Xiao, F. (2018). A non-parametric method to determine basic probability assignment based on kernel density estimation. *IEEE Access*, 6:73509–73519. https://doi.org/10.1109/ACCESS.2018.2883513.

Sen, P. C., Hajra, M., and Ghosh, M. (2020). Supervised classification algorithms in machine learning: A survey and review. In *Emerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018*, pages 99–111. Springer. https://doi.org/DOI: 10.1007/978-981-13-7403-6_11.

Silverman, B. W. (2018). *Density estimation for statistics and data analysis*. Routledge. https://doi.org/10.1201/9781315140919.

Siswanto, S., Mar'ah, Z., Sabir, A. S. D., Hidayat, T., Adhel, F. A., and Amni, W. S. (2022). The sentiment analysis using naïve bayes with lexicon-based feature on tiktok application. *Jurnal Varian*, 6(1):89–96. https://doi.org/10.30812/varian.v6i1.2205.

Tarannum, S. (2023). *Halal Food Identification from Product Ingredients using Machine Learning*. PhD thesis, United International University. https://dspace.uiu.ac.bd/handle/52243/2852.

Weglarczyk, S. (2018). Kernel density estimation and its application. In *ITM web of conferences*, volume 23, page 00037. EDP Sciences.