p–ISSN: 2723 - 6609 e-ISSN: 2745-5254
Vol. 5, No. 5 Mei 2024 http://jist.publikasiindonesia.id/

Classification Of Malaria Types Using Naïve Bayes Classification

Hadin La Ariandi[1*], Arief Setyanto[2], Sudarmawan[3]
Universitas Amikom Yogyakarta, Indonesia
Email: 1293@students.amikom.ac[1*], arief_s@amikom.ac.id[2],
sudarmawan@amikom.ac.id[3]

*Correspondence

Keywords: Naive Bayes Classification; Malaria Type Classification; Expert System for Malaria Diagnosis.

ABSTRACT

This study was conducted to determine the level of accuracy of the naïve Bayes classification method in determining the group type of malaria. This method predicts the malaria category based on the symptoms displayed. This study divided the dataset used into 60% for training and 40% for testing. The results showed that the naïve Bayes algorithm had an accuracy rate of 99.8% in predicting malaria categories. Model performance evaluation using confusion matrix and ROC curve also showed promising results, with classification accuracy of 0.998, error 0.002, and AUC 0.999. The results of the classification report show that the Quartana, Tertiana, and Tropica categories are more dominant than the Ovale categories based on precision, recall, and f1-score. These results show that the naïve Bayes classification method is effective in classifying types of malaria and can be used to diagnose malaria.

Introduction

Malaria is a disease caused by inflammation of protozoa of the genus Plasmodium and is easily recognised by signs of heat, cold, chills, and continuous chills (Dinata, 2018). Malaria is one of the most widespread mosquito-borne diseases (Madhusudan, 2020). Disease caused by inflammation of protozoa from the genus Plasmodium is transmitted through the intermediaries of various vector genera Anopheles (Alviyanil'Izzah et al., 2021). Malaria is still a threat to public health status, especially to people living in remote areas. This is reflected in the issuance of Presidential Regulation Number: 2 of 2015 concerning the National Medium-Term Development Plan for 2015 - 2019, where malaria is a priority disease that needs to be overcome and in RPJMN IV for 2020-2024 it is also stated that the prevalence of major infectious diseases, one of which is malaria is still high accompanied by the threat of emerging diseases due to high population mobility so that it affects the degree of public health (Ramadhan & Khoirunnisa, 2021). This commitment to malaria control is expected to be of concern to all of us nationally, regionally, and globally, as produced at the 60th World Health Assembly (WHA) meeting in Geneva in 2007 on malaria elimination (Prajarini, 2016).

To the World Health Organization (World Health Organization), malaria can be classified into 5, namely plasmodium falciparum, which causes tropical malaria; plasmodium vivax, which causes malaria Persian; plasmodium ovale, which causes maria ovale; plasmodium malaria According According According to causes quaternary malaria, and plasmodium knowlesi causes malaria (Madhusudan, 2020). Malaria is categorised as one of the diseases with effects and a reasonably large mortality rate. The World Health Organization (World Health Organization) recorded 229 million malaria problems and 409. 000 deaths were registered in 2019. Areas at risk are mainly in Africa, but Southeast Asia, the Western Pacific, and the Mediterranean are also listed as areas at risk. Each country strives to overcome malaria cases by referring to the comprehensive commitment in the 60th World Health Assembly (WHA) in 2007 regarding malaria elimination (Jiang et al., 2021).

The objectives of this study are:

1. Knowing the level of accuracy of the naïve Bayes classification method in determining the group of types of malaria.
2. Knowing how many results are accurate and the performance of malaria types using the naïve Bayes algorithm.
3. Prove whether the naïve Bayes classification method effectively classifies malaria types.

Research Benefits

With the research that will be held, several hopes for the results of this research can be helpful and play an essential role in adding insight into science. The benefits obtained by conducting this research are as follows:

1. Mitigating and assisting the performance of medical professionals in classifying types of malaria.
2. Provide information on the level of accuracy in the process of classifying malaria.
3. Adding insight for readers who want to learn naïve Bayes classification.

Research Methods

Researchers use quantitative research, a process of mathematical calculations, to achieve the desired results. In this case, the dataset was compared with the Naïve Bayes algorithm to find the most malaria-related impacts in each Puskesmas in Irian Jaya.

Nature of Research

The nature of the research carried out is experimental. It conducts a research experiment to obtain accurate results or parameters by comparing the Naïve Bayes algorithm. The accuracy results obtained from the comparison can be used to make decisions about determining the feasibility of lending.

Research Approach

This research approach is quantitative, and researchers conduct research by the stages or lines of research that have been made.

Data Collection Methods

The data used in this study is obtained directly from the Darun Nahdla Capita Sharia Cooperative and includes private data that has not been used in previous studies. The data used in this study is from datasets from cooperative customer data from 2020 to 2022, totalling 166 data points with 10 variables: gender, marital status, occupation, dependents, income, loan amount, term, interest, instalments, and categories.

Data Analysis Methods

The data analysis method for this study is quantitative, while the data analysis method follows the stages in the knowledge discovery in database (kdd) process used in this study using Excel software tools and orange tools as follows:

Research Flow

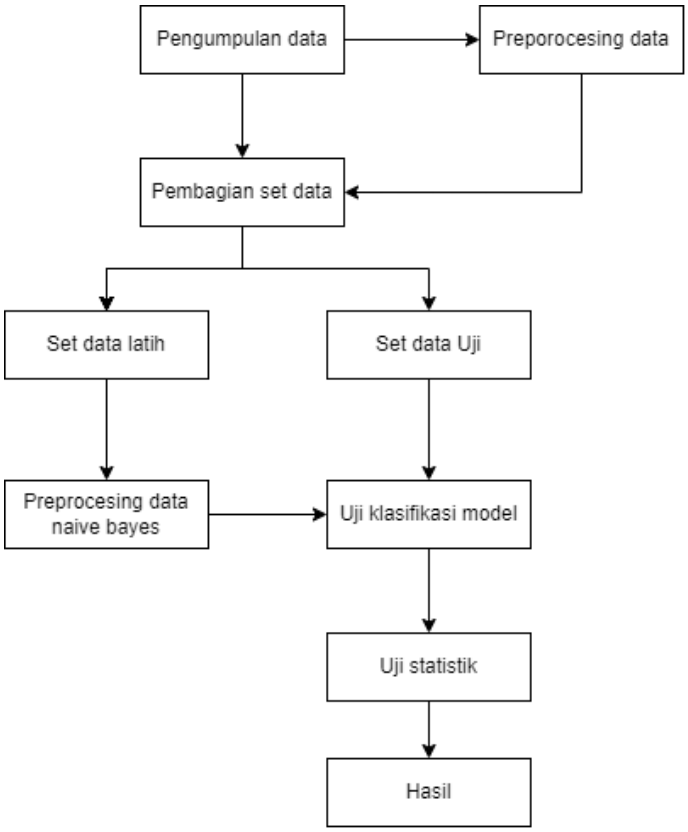

Figure 1
Research Flow

Results and Discussion

Preprocessing Data

The data preprocessing stage is carried out to clean duplicate data, missing values, and outliers in the dataset so that they are valid during the data processing. At this stage, data transformation is also carried out by analysing variables that do not have contributive information to make predictions and converting object-type data into integer form to facilitate the data processing process. The following data preprocessing process uses Jupyter Notebook software with Python programming language (Lestari et al., 2018).

The first step is to import the library that will be used to display the dataset using the numpy and Pandas methods, which can be seen in the code below.

```
import numpy as np
import pandas as PD
import matplotlib.pyplot as plt
import seaborn as sns
```

The second step is to call the CSV format dataset into the data frame with the PD.read_csv function and display the dataset, code and output results, as shown in Figure 2 below.

```
filecsv='Dataset_Patient_Malaria.CSV
teks = pd.read_csv(files, header = 0, delimiter= ';', encoding='utf-8')
df=pd.DataFrame(teks)
print(df)
df.head()
```

output:

No.	Provinsi	Kabupaten	Fasyankes	Jenis Penemuan	Nama Pasien	Angka	Bulan / Tahun	Jenis Kelamin	Hamil / Tidak Hamil	...	Genangan_Air	Riwayat tinggal_di daerah endemis
1	PAPUA	KOTA JAYAPURA	PUSKESMAS KOYA BARAT	Passive Case Detection	AURISTA RUSSELL CHYIANTHIKA ANTHONY	13	Tahun	P	Tidak Hamil	...	Tidak Ada	Ya
2	PAPUA	KOTA JAYAPURA	PUSKESMAS KOYA BARAT	Passive Case Detection	MUHAMAD FAIQ BADRUL SHOLEH	13	Tahun	L	Tidak Hamil	...	Tidak Ada	Ya
3	PAPUA	KOTA JAYAPURA	PUSKESMAS KOYA BARAT	Passive Case Detection	NORCE MANTOL	36	Tahun	P	Tidak Hamil	...	Tidak Ada	Ya
4	PAPUA	KOTA JAYAPURA	PUSKESMAS KOYA BARAT	Passive Case Detection	PAULINA ROKKI	28	Tahun	P	Tidak Hamil	...	Tidak Ada	Ya
5	PAPUA	KOTA JAYAPURA	PUSKESMAS KOYA BARAT	Passive Case Detection	KARYADI	31	Tahun	L	Tidak Hamil	...	Tidak Ada	Ya

ws × 37 columns

Figure 2 Import Research Dataset

Figure 2 shows the 37 dataset variables used in this study, and several are unnecessary, such as No, province, district, health facility, and patient name.

The third step deletes the columns not needed for the next process and the columns to be deleted.

```
columns = ['No.','Provinsi ', 'Kabupaten','Fasyankes','Nama Pasien']
copy = df
dfClean = dfCopy.drop(columns, inplace=True, axis=1)
list(df.columns)
```

After deleting the columns that are not needed, the following columns will be used for the following process: type of discovery, number, month/year, gender, pregnant / not pregnant, hamlet address, village kelurahan, type of parasite, symptoms1, symptoms2,

symptoms3, symptoms4, symptoms5, symptoms6, symptoms7, symptoms8, symptoms9, symptoms10, livestock sheds, leaving the house at night, use of mosquito repellent, ventilation gauze, puddles, history of living in endemic areas, the use of mosquito nets, walls, the state of the house sky, mosquito breeding grounds, air temperature (°C), humidity (%), rainfall (mm), malaria diagnosis (Shofia, Putri, & Arwan, 2017).

The fourth step separates variables into category and number variables using the following code command:

#untuk define category variables

categorical = [var for var in pdf. columns if df[var].dtype=='O']

Output:

Discovery Type', 'Month/Year', 'Gender', 'Pregnant/Not Pregnant', 'Dusun_Alamat', 'Village Village', 'Parasite Type', 'Symptoms1', 'Symptoms2', 'Symptoms3', 'Symptoms4', 'Symptoms5', 'Symptoms6', 'Symptoms7', 'Symptoms8', 'Symptoms9', 'Symptoms10', 'Kandang_Ternak', 'Night rumah_pada Exit', 'Mosquito Obat_Anti Use', 'Kassa_Ventilasi', 'Genangan_Air', 'History of tinggal_di endemic areas', 'Penggunaan_Kelambu', 'Walls', 'House sky conditions', 'Mosquito Breeding Sites', 'Diagnosa_Malaria']

#to define a number variable

numerical = [var for var in pdf.columns if df[var].dtype!='O']

output:

['Number', 'Air Temperature (°C)', 'Humidity (%)', 'Rainfall (mm)']

Next, do data cleaning to clean up duplicate data or unused variables, missing values and outliers. The code and output results can be seen in Figure 3 below.

df[categorical].isnull().sum()

df[numerical].isnull().sum()

```
Jenis Penemuan                      0
Bulan / Tahun                       0
Jenis Kelamin                       0
Hamil / Tidak Hamil                 0
Dusun_Alamat                        0
Desa Kelurahan                      0
Jenis Parasit                       0
Gejala1                             0
Gejala2                             0
Gejala3                             0
Gejala4                            69
Gejala5                           107
Gejala6                           108
Gejala7                             1
Gejala8                             3
Gejala9                             1
Gejala10                            1
Kandang_Ternak                      0
Keluar rumah_pada malam hari        0
Penggunaan Obat_Anti Nyamuk         0
Kassa_Ventilasi                     0
Genangan_Air                        0
Riwayat tinggal_di daerah endemis   0
Penggunaan_Kelambu                  0
Dinding                             0
Keadaan langit rumah                0
Tempat Perindukan Nyamuk            0
Diagnosa_Malaria                    0
dtype: int64
```

Figure 3
Check the Dataset missing value category variable.

In Figure 3, the results above show that no null values are used in the dataset of category variables other than symptom variables because symptoms can be empty (only some symptoms).

```
Angka              0
Suhu Udara (°C)    0
Kelembaban (%)     0
Curah Hujan (mm)   0
dtype: int64
```

Figure 4
Check the Dataset missing value variable number.

The result above in Figure 4 shows that no null values are used in the numeric variable dataset. Each column has the same number of null values as zero. With no null values other than symptom variables for category variables, this dataset appears to be pretty clean and does not require any special steps to handle missing values (Fajar et al., 2018).

Next, define the dependent and independent variables on the dataset. The dependent variables selected are type of discovery, number, month/year, gender, pregnant / not pregnant, hamlet address, village kelurahan, type of parasite, symptom1, symptom2, symptom3, symptom4, symptom5, symptom6, symptom7, symptom8, symptom9, symptom10, livestock shed, leaving the house at night, use of mosquito repellent,

ventilation gauze, puddles, history of living in endemic areas, use of mosquito nets, walls, state of the house sky, mosquito breeding site, air temperature (°C), humidity (%), rainfall (mm) as independent variables with the ILOC method to select dependent and independent variables based on column/variable index. In this case, it will use x, which contains all dependent variables, and y, which contains the independent or target variable. The code and output results can be seen in Figures 5 and 6 below.

#Menentukan dependent and independent variables

X = df.drop(['Diagnosa_Malaria'], axis=1)

y = df['Diagnosa_Malaria']

#Display dependent variables and independent variables

print (X)

print (y)

Output x:

```
           Jenis Penemuan  Angka Bulan / Tahun Jenis Kelamin  \
0  Passive Case Detection     13         Tahun             P
1  Passive Case Detection     13         Tahun             L
2  Passive Case Detection     36         Tahun             P
3  Passive Case Detection     28         Tahun             P
4  Passive Case Detection     31         Tahun             L

  Hamil / Tidak Hamil              Dusun_Alamat          Desa Kelurahan   \
0         Tidak Hamil  Asei Kecil Sentani Timur  Dalam Wilayah Kabupaten
1         Tidak Hamil               Heram Waena  Dalam Wilayah Kabupaten
2         Tidak Hamil       Tablanusu Tablasupa   Luar Wilayah Kabupaten
3         Tidak Hamil               Heram Hedam  Dalam Wilayah Kabupaten
4         Tidak Hamil               Heram Waena  Dalam Wilayah Kabupaten

           Jenis Parasit Gejala1    Gejala2  ... Kassa_Ventilasi Genangan_Air  \
0       Plasmodium vivax   Demam  menggigil  ...  Tidak Tersedia    Tidak Ada
1       Plasmodium vivax   Demam  menggigil  ...         Tersedia    Tidak Ada
2       Plasmodium vivax   Demam  menggigil  ...  Tidak Tersedia    Tidak Ada
3  Plasmodium falciparum   Demam  menggigil  ...  Tidak Tersedia    Tidak Ada
4       Plasmodium vivax   Demam  menggigil  ...  Tidak Tersedia    Tidak Ada

  Riwayat tinggal_di daerah endemis Penggunaan_Kelambu      Dinding  \
0                                Ya                 Ya        Rapat
1                                Ya                 Ya        Rapat
2                                Ya                 Ya  Tidak Rapat
...
2            76             0.0
3            90             0.0
4            90             0.0

[5 rows x 31 columns]
```

Figure 5 Dependent Variables

Output y:

```
0       Tertiana
1       Tertiana
2       Tertiana
3        Tropica
4       Tertiana
          ...
1279     Tropica
1280    Quartana
1281     Tropica
1282    Tertiana
1283    Tertiana
Name: Diagnosa_Malaria, Length: 1284, dtype: object
```

Figure 6 Independent Variables

The output above shows that the dependent variable (X) consists of 31 variables for the independent variable (Y), namely the malaria diagnosis.

1. Correlation of the independent variable to the dependent variable

The correlation of the dependent variable to the independent variable is carried out to determine how much influence the dependent variable/predictor has on the independent / target variable (Shofia et al., 2017). The correlation of independent variables based on the dependent variable/predictor can be seen in Table 1 below.

Tabel 1
Korelasi Antar Variabel

No	Variable	Result
1.	Types of Inventions	-
2.	Month / Year	-0.043122
3.	Gender	-0.045759
4.	Pregnant / Not Pregnant	0.013941
5.	Hamlet Address	-0.014255
6.	Village Village	-0.02373
7.	Types of parasites	0.246237
8.	Gejala1	-
9.	Gejala2	-0.071117
10.	Gejala3	0.208725
11.	Gejala4	0.037009
12.	Gejala5	-0.078445
13.	Gejala6	0.038522
14.	Gejala7	-0.127901
15.	Gejala8	0.121262
16.	Gejala9	-0.23095
17.	Gejala10	-0.656569
18.	Cattle shed	-0.03218
19.	Go out at night	-0.016352
20.	Use of mosquito repellent	-0.027067
21.	Cashier Ventilasi	-0.002299
22.	Puddle	-0.031197
23.	History of living in endemic areas	-0.016352

24.	Use of mosquito nets	-0.016352
25.	Wall	0.041548
26.	The state of the house's sky	0.02998
27.	Mosquito Breeding Place	-0.03218
28.	Angka	0.024472
29.	Air Temperature (°C)	0.075598
30.	Humidity (%)	0.050986
31.	Precipitation (mm)	-0.02056

Based on Table 1 above, it can be seen that the variables of discovery type, month/year, gender, hamlet Address, village, symptom 1, symptom 2, symptom 5, symptom 7, symptom 9, symptom 10, livestock drums, leaving the house at night, use of mosquito repellent, ventilation gauze, puddles, history of living in endemic areas, use of mosquito nets, mosquito breeding sites and rainfall (mm) do not affect the dependent variable or target variable. Based on the calculation results, the correlation value obtained is negative, so it can be said that the variable does not strongly influence the dependent variable or target (Setiawan & Prihandono, 2019).

Model Testing

The model used to perform testing on the research dataset is the naïve Bayes algorithm model. Model testing is performed to display the classification report of the model used to see the value of classification evaluation metrics such as precision, recall, F1-score, and accuracy.

Naïve bayes Algorithm Model Testing

Testing on datasets is carried out using the Naïve Bayes algorithm to determine the classification report and accuracy in making classifications or predictions. The following testing process uses Jupyter Notebook software with Python programming language.

Testing the naïve Bayes algorithm with split or 90/10 data sharing for code and output results can be seen below.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.10, random_state = 0)
y_pred = gnb.predict(X_test)
from sklearn import metrics
from sklearn.metrics import classification_report
cr1 = classification_report(y_test, y_pred)
akurasi = metrics.accuracy_score(y_test, y_pred)
rint (cr1)
print ('The value of accuracy possessed by the model: %0.2f ' %(akurasi*100),'%')
```

```
precision    recall f1-score   support
Ovale       1.00      1.00      1.00         5
Quartana    1.00      0.86      0.92         7
Tertiana    0.98      1.00      0.99        61
Tropica     1.00      1.00      1.00        56
```

accuracy			0.99	129
macro avg	1.00	0.96	0.98	129
weighted avg	0.99	0.99	0.99	129

Accuracy value possessed by the model: 99.22 %

The above results can be explained. Precision is the ratio of correctly predicted positive observations to predicted positive totals. The precision for the Ovale class is 1.00, which means all class data predicted as the Ovale class is correct. The precision for the Quartana class is 1.00, which means all class data predicted as the Quartana class is correct. The precision for the Tertiana class is 0.98, which means that 98% of the class data predicted as the Tertiana class is the Tertiana class. The precision for the Tropica class is 1.00, which means all class data predicted as the Tropica class is correct. Recall is the ratio of correctly predicted positive observations to all actual positives. The recall for the Ovale, Quartana, and Tropica classes is 1.00, indicating that the model correctly identifies all instances of those classes. The recall for the Tertiana class is 0.98, which means the model manages to capture 98% of the actual instances of the Tertiana class. The F1-Score is a weighted average of precision and recall. The range is from 0 to 1, where 1 is the best F1-Score. The F1-Score for the Ovale and Tropica classes is 0.97, reflecting a good balance between precision and recall for the Ovale and Tropica classes. The F1-score for the Quartana class is 0.92, and the Tertiana class is 0.99, indicating a somewhat lower balance between precision and recall for the Quartana class Tertiana class compared to the Ovale class and Tropica class. Support indicates the actual number of class occurrences in the specified dataset. There are 5 Ovale class data, 7 Quartana class data, 61 Tertiana class data and 56 Tropica class data. The overall accuracy is 99.22%, representing the ratio of correctly predicted class data to total class data. Overall, the model performs well, especially for Ovale-class, Tertiana-class and Tropica-class data, achieving high precision and recall. For the Quartana class, the precision is perfect, but the recall is slightly lower, showing some difficulty in capturing all the data for the Quartana class (Shen & Shafiq, 2020).

Testing the naïve Bayes algorithm with split or 80/20 data division for code and output results can be seen below.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)
y_pred = gnb.predict(X_test)
from sklearn import metrics
from sklearn.metrics import classification_report
cr1 = classification_report(y_test, y_pred)
akurasi = metrics.accuracy_score(y_test, y_pred)
print(cr1)
print ('The accuracy value possessed by the model: %0.2f ' %(akurasi*100),'%')
```

	precision	recall	f1-score	support
Ovale	0.95	1.00	0.97	18
Quartana	1.00	1.00	1.00	15
Tertiana	1.00	0.99	1.00	118
Tropica	1.00	1.00	1.00	106
accuracy		1.00		257
macro avg	0.99	1.00	0.99	257
weighted avg	1.00	1.00	1.00	257

Accuracy value owned by the model: 99.61%

Based on the results above, 80% of training and 20% of testing data sharing can be explained. The precision for an Ovale class is 0.95, which means that 95% of the class data predicted to be an Ovale class is an Ovale class. The precision for the Quartana, Tertiana, and Tropica classes is 1.00, meaning all class data is predicted as correct. The recall for the Ovale, Quartana, and Tropica classes is 1.00, indicating that the model correctly identifies all instances of those classes. The recall for the Tertiana class is 0.99, which means the model captures 99% of the actual instances of the Tertiana class. The F1-Score is a weighted average of precision and recall. The range is from 0 to 1, where 1 is the best F1-Score. The F1-Score for the Quartana, Tertiana and Tropica classes is 1.00, reflecting a good balance between precision and recall for the Quartana Tropica and Tertiana classes. The F1-score for the Ovale class is 0.97, indicating a somewhat lower balance between precision and recall for the Ovale class compared to the Quartana Tropica and Tertiana classes. Support indicates the actual number of class occurrences in the specified dataset. There are 18 Ovale class data, 15 Quartana class data, 118 Tertiana class data and 106 Tropica class data. The overall accuracy is 99.61%, representing the ratio of correctly predicted class data to total class data. Overall, the model performs well, especially for Tropica-class and Quartana-class data, achieving high precision and recall. For the Tertiana class, the precision is perfect, but the recall is slightly lower, showing some difficulty in capturing all the data of the Tertiana class.

Testing the naïve Bayes algorithm with split or 70/30 data division for code and output results can be seen below.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30, random_state = 0)
y_pred = gnb.predict(X_test)
from sklearn import metrics
from sklearn.metrics import classification_report
cr1 = classification_report(y_test, y_pred)akurasi = metrics.accuracy_score(y_test, y_pred)
print(cr1)
print ('Nilai akurasi yang dimiliki oleh model: %0.2f ' %(akurasi*100),'%')
```

	precision	recall	f1-score	support
Ovale	0.96	1.00	0.98	27
Quartana	1.00	1.00	1.00	18
Tertiana	1.00	0.99	1.00	178
Tropica	1.00	1.00	1.00	163
accuracy			1.00	386
macro avg	0.99	1.00	0.99	386
weighted avg	1.00	1.00	1.00	386

Accuracy value possessed by the model: 99.74%

Based on the results above, 70% of training and 30% of testing data sharing can be explained. The precision for an Ovale class is 0.96, which means 96% of the class data predicted as an Ovale class is an Ovale class. The precision for the Quartana, Tertiana, and Tropica classes is 1.00, meaning all class data is predicted as correct. The recall for the Quartana and Tropica classes is 1.00, indicating that the model correctly identifies all instances of those classes. The recall for the Tertiana class is 0.99, which means the model captures 99% of the actual instances of the Tertiana class. The F1-Score is a weighted average of precision and recall. The range is from 0 to 1, where 1 is the best F1-Score. The F1-Score for the Quartana, Tertiana and Tropica classes is 1.00, reflecting a good balance between precision and recall for the Quartana, Tertiana and Tropica classes. Support indicates the actual number of class occurrences in the specified dataset. There are 27 Ovale class data, 18 Quartana class data, 178 Tertiana class data and 163 Tropica class data. The overall accuracy is 99.74%, representing the ratio of correctly predicted class data to total class data. The model performs well, especially for Quartana and Tropica class data, where high precision and recall are achieved. For the Tertiana class, the precision is perfect, but the recall is slightly lower, showing some difficulty in capturing all the data of the Tertiana class.

Testing the naïve Bayes algorithm with split or 60/40 data division for code and output results can be seen below.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.40, random_state = 0)
y_pred = gnb.predict(X_test)
from sklearn import metrics
from sklearn.metrics import classification_report
cr1 = classification_report(y_test, y_pred)
akurasi = metrics.accuracy_score(y_test, y_pred)
print(cr1)
print('The value of accuracy possessed by the model: %0.2f ' %(akurasi*100),'%')
```

	precision	recall	f1-score	support
Ovale	0.97	1.00	0.99	36

Quartana	1.00	1.00	1.00	23
Tertiana	1.00	1.00	1.00	242
Tropica	1.00	1.00	1.00	213
accuracy		1.00		514
macro avg	0.99	1.00	1.00	514
weighted avg	1.00	1.00	1.00	514

The accuracy value possessed by the model is 99.81 %

Based on the above results, 60% training and 40% testing can be explained with data sharing. The precision for an Ovale class is 0.97, which means 97% of the class data predicted as an Ovale class is an Ovale class. The precision for the Quartana, Tertiana, and Tropica classes is 1.00, meaning all class data is predicted as correct. The recall for classes Ovale, Quartana, Tertiana, and Tropica is 1.00, indicating that the model correctly identifies all instances of those classes. The F1-Score is a weighted average of precision and recall. The range is from 0 to 1, where 1 is the best F1-Score. The F1-Score for the Quartana, Tertiana and Tropica classes is 1.00, reflecting a good balance between precision and recall for the Quartana, Tertiana and Tropica classes. Support indicates the actual number of class occurrences in the specified dataset. There are 36 Ovale class data, 23 Quartana class data, 242 Tertiana class data and 213 Tropica class data. The overall accuracy is 99.81%, representing the ratio of correctly predicted class data to total class data. Overall, the model performs well, especially for the Quartana-class, Tertiana-class and Tropica-class data, achieving high precision and recall.

Testing the naïve Bayes algorithm with split or 50/50 data division for code and output results can be seen below.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.50, random_state = 0)
y_pred = gnb.predict(X_test)
from sklearn import metrics
from sklearn.metrics import classification_report
cr1 = classification_report(y_test, y_pred)
akurasi = metrics.accuracy_score(y_test, y_pred)
print(cr1)
print ('The value of accuracy possessed by the model: %0.2f ' %(akurasi*100),'%')
```

	precision	recall	f1-score	support
Ovale	0.98	1.00	0.99	44
Quartana	1.00	1.00	1.00	31
Tertiana	1.00	1.00	1.00	299
Tropica	1.00	1.00	1.00	268
accuracy		1.00		642
macro avg	0.99	1.00	1.00	642

weighted avg 1.00 1.00 1.00 642
Accuracy value possessed by the model: 99.84%

The data sharing of 50% training and 50% testing can be explained based on the results above. The precision for an Ovale class is 0.98, which means that 98% of the class data predicted as an Ovale class is an Ovale class. The precision for the Quartana, Tertiana, and Tropica classes is 1.00, meaning all class data is predicted as correct. The recall for classes Ovale, Quartana, Tertiana, and Tropica is 1.00, indicating that the model correctly identifies all instances of those classes. The F1-Score is a weighted average of precision and recall. The range is from 0 to 1, where 1 is the best F1-Score. The F1-Score for the Quartana, Tertiana and Tropica classes is 1.00, reflecting a good balance between precision and recall for the Quartana, Tertiana and Tropica classes. Support indicates the actual number of class occurrences in the specified dataset. There are 44 Ovale class data, 31 Quartana class data, 299 Tertiana class data and 268 Tropica class data. The overall accuracy is 99.84%, representing the ratio of correctly predicted class data to total class data. Overall, the model performs well, especially for the Quartana-class, Tertiana-class and Tropica-class data, achieving high precision and recall.

Based on the classification results of the Naïve Bayes algorithm, it can be concluded that the results of the classification report on the algorithm show that the Quartana, Tertiana and Tropica categories are more dominant than the Ovale category because the precision, recall and f1-score values in the Quartana, Tertiana and Tropica categories are higher than the precision, recall and f1-score values in the Ovale category. Then, the highest accuracy value was obtained by the naïve Bayes algorithm in the fifth test with a 50/50 data division of 99.84%. More details can be seen in Table 2 below.

Table 2
Classification Report Naïve Bayes

Algoritma Klasifikasi	Category	Precision	Recall	F1-Score	Support	Accuracy
Naïve Bayes (90/10)	Oval	1.00	1.00	1.00	5	99.22%
	Quartana	1.00	0.86	0.92	7	
	Tertiana	0.98	1.00	0.99	61	
	Tropica	1.00	1.00	1.00	56	
Naïve Bayes (80/20)	Oval	0.95	1.00	0.97	18	99.61%
	Quartana	1.00	1.00	0.92	15	
	Tertiana	1.00	0.99	1.00	118	
	Tropica	1.00	1.00	1.00	106	
Naïve Bayes (70/30)	Oval	0.96	1.00	0.98	27	99.74%
	Quartana	1.00	1.00	1.00	18	
	Tertiana	1.00	0.99	1.00	178	
	Tropica	1.00	1.00	1.00	163	
Naïve Bayes (60/40)	Oval	0.97	1.00	0.99	36	99.81%
	Quartana	1.00	1.00	1.00	23	
	Tertiana	1.00	1.00	1.00	242	
	Tropica	1.00	1.00	1.00	213	

Naïve Bayes (50/50)	Oval	0.98	1.00	0.99	44	99.84%
	Quartana	1.00	1.00	1.00	31	
	Tertiana	1.00	1.00	1.00	299	
	Tropica	1.00	1.00	1.00	268	

In Table 2 above, it can be seen that the highest value obtained by the naïve Bayes algorithm in the fifth test, whose accuracy value was 99.84%, with a 50/50 data division.

Evaluation

At this stage, the Naïve Bayes algorithm was evaluated using the Confusion Matrix method and the Receiver Operating Characteristic (ROC) curve. To find out the model's performance on each algorithm with the help of jupyter notebook software Python programming language.

Based on the results of the confusion matrix model evaluation, it can be seen that the performance accuracy of the naïve Bayes algorithm model is 0.992, and the classification error is 0.008. Furthermore, evaluation of the naïve Bayes algorithm model was carried out using ROC to visually measure the performance of the classification model, focusing on True Positive Rate and False Positive Rate at one point to provide information on the performance of the naïve Bayes algorithm model in general.

Based on the figure above, the evaluation results of the naïve Bayes algorithm, which compares the performance of data classification with the Area Under Curve (AUC) technique of 0.976, are included in the excellent classification.

Based on the results of the confusion matrix model evaluation, it can be seen that the performance accuracy of the naïve Bayes algorithm model is 0.998, and the classification error is 0.002. Furthermore, an evaluation of the naïve Bayes algorithm model was carried out using the ROC curve to visually measure the performance of the classification model, focusing on the True Positive Rate and False Positive Rate at one point to be able to provide information on the performance of the naïve Bayes algorithm model in general.

Table 3 below shows the results of the performance evaluation of the Naïve Bayes algorithm model using the confusion matrix and the ROC curve.

Tabel 3
Evaluasi Confusion Matrix dan Kurva ROC Naïve Bayes

Evaluation Algoritma	Confusion Matrix		Fucking ROC
	Classification Accuracy	Classification errors	AUC
Naïve Bayes (90/10)	0.992	0.008	0.976
Naïve Bayes (80/20)	0.996	0.004	0.999
Naïve Bayes (70/30)	0.997	0.003	0.999
Naïve Bayes (60/40)	0.998	0.002	0.999
Naïve Bayes (50/50)	0.998	0.002	0.999

Best Results

Based on the results of data processing using jupyter notebook software using the Python programming language on the naïve Bayes algorithm in classifying or predicting the Tertiana, Tropica, Quartana and Ovale categories in malaria diagnosis, it is known that the naïve Bayes algorithm with 60/40 and 50/50 dataset division evaluation has the highest accuracy rate with an accuracy of 99.8%. Furthermore, in the evaluation of the performance model using the confusion matrix, a classification accuracy of 0.998 and a classification error of 0.002 was obtained, then an evaluation using the ROC curve that focuses on True Positive Rate and False Positive Rate at one point to be able to provide general algorithm performance information with an AUC of 0.999.

Conclusion

Based on the results of research on malaria diagnosis with the algorithm used, namely Naïve Bayes, conclusions can be drawn:

1. The classification results of the Naïve Bayes algorithm have an accuracy of 99.8%
2. The performance evaluation of the confusion matrix model and the ROC curve of the Naïve Bayes algorithm has a classification accuracy of 0.998, an error of 0.002, and an AUC of 0.999.
3. The results of the classification report from the algorithm show that the Quartana, Tertiana and Tropica categories are more dominant than the categories because the precision, recall and f1-score values in the Quartana, Tertiana and Tropica categories are higher than the precision, recall and f1-score values in the Ovale category.

Bibliography

Alviyanil'Izzah, Nur, Martia, Dina Yeni, Imaculata, Maria, Hidayatullah, Moh Iqbal, Pradana, Andhika Bagus, Setiyani, Diyah Ayu, & Sapuri, Enes. (2021). Analisis Teknikal Pergerakan Harga Saham Dengan Menggunakan Indikator Stochastic Oscillator Dan Weighted Moving Average. *Keunis*, *9*(1), 36–53. https://doi.org/10.32497/keunis.v9i1.2307

Dinata, A. (2018). *Bersahabat dengan Nyamuk: Jurus Jitu Atasi Penyakit Bersumber Nyamuk*. Arda Publishing House.

Fajar, Riyant, Perdana, Rizal Setya, & Indriati, Indriati. (2018). Implementasi Metode Naïve Bayes Dengan Perbaikan Missing Value Menggunakan Metode Nearest Neighbor Imputation Studi Kasus: Penyakit Malaria Di Kabupaten Malang. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *2*(8), 2430–2434.

Jiang, G., Liu, Fen, Liu, Wenping, Shan, Chen, Yufeng, & Xu, Dongming. (2021). Effects of information quality on information adoption on social media review platforms: The moderating role of perceived risk. *Data Science and Management*, *1*(1), 13–22.

Lestari, Indri Dwi, Setiadi, Tedy, & Zahrotun, Lisna. (2018). Penerapan Data Mining Menggunakan Metode Naïve Bayes Untuk Klasifikasi Tindakan Jenis Abortus Di Rsud Duta Mulya. *Jurnal Sarjana Teknik Informatika*, *6*(2), 60–68.

Madhusudan, Desai Mitesh. (2020). Stock Closing Price Prediction Using Machine Learning SVM Model. *International Journal for Research in Applied Science and Engineering Technology*.

Prajarini, D. (2016). Perbandingan Algoritma Klasifikasi Data Mining Untuk Prediksi Penyakit Kulit. *INFORMAL: Informatics Journal*, *1*(3), 137–141.

Ramadhan, Nur Ghaniaviyanto, & Khoirunnisa, Azka. (2021). Klasifikasi Data Malaria Menggunakan Metode Support Vector Machine. *Jurnal Media Informatika Budidarma*, *5*(4), 1580–1584. https://doi.org/10.30865/mib.v5i4.3347

Setiawan, Aries, & Prihandono, Adi. (2019). Klasifikasi Tingkat Kerentanan Malaria Pada Suatu Wilayah Menggunakan Na ৲ Ve Bayes Data Mining. *VISIKES: Jurnal Kesehatan Masyarakat*, *18*(1).

Shen, Jingyi, & Shafiq, M. Omair. (2020). Short-term stock market price trend prediction using a comprehensive deep learning system. *Journal of Big Data*, *7*, 1–33.

Shofia, Elsa Nuramilus, Putri, Rekyan Regasari Mardi, & Arwan, Achmad. (2017). Sistem Pakar Diagnosis Penyakit Demam: DBD, Malaria dan Tifoid Menggunakan Metode K-Nearest Neighbor-Certainty Factor. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *1*(5), 426–435.