

Tugas 9: Implementasi Algoritma Naive Bayes untuk Klasifikasi Dataset Cancer

Monika Septiana - 0110222127 ^{1*}

¹ Teknik Informatika, STT Terpadu Nurul Fikri, Depok

*E-mail: moni22127ti@student.nurulfikri.ac.id

Abstract. Penelitian ini membahas penerapan algoritma Naive Bayes untuk klasifikasi data kanker payudara menggunakan dataset dari Kaggle. Proses dimulai dengan melakukan preprocessing data, pemisahan fitur dan label, serta pembagian data menjadi training dan testing. Model Naive Bayes kemudian dilatih dan diuji untuk mengukur performa klasifikasi. Hasil evaluasi menunjukkan bahwa algoritma ini mampu memberikan akurasi tinggi dalam membedakan antara kategori *Malignant* dan *Benign*.

Naive Bayes pada Dataset Cancer

Setup & Import Library

```
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```
# Praktikum Mandiri - Naive Bayes pada Dataset Cancer
```

```
# Import library yang dibutuhkan
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import seaborn as sns
import matplotlib.pyplot as plt
```

Load Dataset

Pastikan file data.csv dari Kaggle sudah diupload ke Colab (menu: Files → Upload).

```
# Load dataset
data = pd.read_csv("/content/drive/MyDrive/praktikum_ml/praktikum09/data/data.csv")

# Tampilkan 5 baris pertama
data.head()
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_m
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07884	0.0
2	84300803	M	19.89	21.25	130.00	1203.0	0.10960	0.15990	0.1
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1

5 rows × 10 columns

Preprocessing

```
# Hapus kolom yang tidak relevan
data = data.drop(['id', 'Unnamed: 32'], axis=1)

# Pisahkan fitur (X) dan label (y)
X = data.drop('diagnosis', axis=1)
y = data['diagnosis']
```

Split Data

```
# Split data menjadi training dan testing
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
)
```

Training Model Naive Bayes

```
# Inisialisasi model Gaussian Naive Bayes
model = GaussianNB()
```

```
# Training
model.fit(X_train, y_train)

# Prediksi
y_pred = model.predict(X_test)
```

✓ Evaluasi Model

```
# Akurasi
print("Accuracy:", accuracy_score(y_test, y_pred))

# Confusion Matrix
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))

# Classification Report
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

Accuracy: 0.9415204678362573

Confusion Matrix:
[[104 4]
[6 57]]

Classification Report:		precision	recall	f1-score	support
B	0.95	0.96	0.95	108	
M	0.93	0.90	0.92	63	
accuracy			0.94	171	
macro avg	0.94	0.93	0.94	171	
weighted avg	0.94	0.94	0.94	171	

✓ Visualisasi Confusion Matrix

```
plt.figure(figsize=(6,4))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues',
            xticklabels=['Benign', 'Malignant'],
            yticklabels=['Benign', 'Malignant'])
plt.title("Confusion Matrix - Naive Bayes")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

