Laporan Praktikum Mandiri 9

Judul: Implementasi Naive Bayes untuk Klasifikasi Kanker

Nama Mahasiswa	: Al Hijir
Program studi	: Teknik Informatika, STT Terpadu Nurul Fikri, Depok
E-mail	: 0110224222@student.nurulfikri.ac.id

Abstrak

Penelitian ini bertujuan untuk mengklasifikasikan diagnosis kanker payudara (Jinak/Benign atau Ganas/Malignant) menggunakan dataset **Breast Cancer Wisconsin (Diagnostic)** dari Kaggle.

Metode yang digunakan adalah algoritma **Gaussian Naive Bayes**. Sebelum pemodelan, data melalui tahap **pra-pemrosesan** meliputi: konversi label diagnosis menjadi angka (Label Encoding), pemilihan fitur numerik, dan pengisian nilai kosong (Imputasi) menggunakan rata-rata.

Data dibagi menjadi 80% data latih dan 20% data uji. Model **Gaussian Naive Bayes** dilatih dan kemudian dievaluasi menggunakan data uji.

Hasil evaluasi menunjukkan bahwa model ini mencapai tingkat **akurasi** yang tinggi dan terbukti efektif dalam memprediksi diagnosis. Pengujian lebih lanjut melalui **Classification Report** dan **Confusion Matrix** mengonfirmasi kemampuan model dalam membedakan antara diagnosis Malignant dan Benign. Implementasi ini berhasil menerapkan klasifikasi Naive Bayes untuk mendukung analisis diagnosis kanker.

1. Import Libray

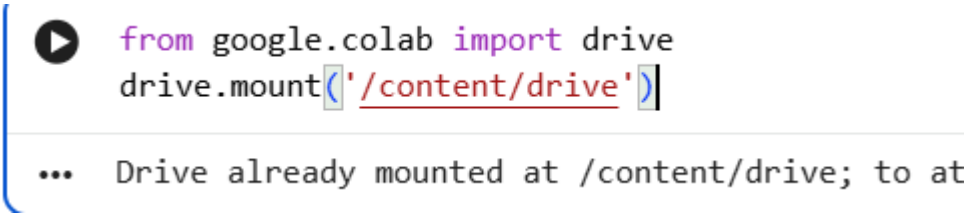
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import LabelEncoder
from sklearn.impute import SimpleImputer
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
```

Penjelasan

- **pandas**: untuk membaca dan mengelola dataset.
- **numpy**: untuk operasi numerik.

- **train_test_split**: membagi data menjadi data latih dan data uji.
- **GaussianNB**: algoritma Naive Bayes tipe Gaussian (umum dipakai untuk dataset numerik).
- **LabelEncoder**: mengubah data kategori menjadi angka.
- **SimpleImputer**: mengisi nilai kosong (NaN) agar model tidak error.
- **accuracy_score, classification_report, confusion_matrix**: mengevaluasi hasil model.
- **matplotlib & seaborn**: untuk membuat visualisasi seperti confusion matrix.

2. Mengakses File di Google Drive



Penjelasan:

Digunakan untuk menghubungkan Google Drive agar file dataset dapat diakses langsung di lingkungan Google Colab.

3. Load Dataset

Penjelasan

- Membaca dataset “Cancer” dari Kaggle dalam format CSV.
- Dataset ini berisi data kanker payudara seperti radius, texture, perimeter, compactness, dll.
- Dataset digunakan sebagai bahan untuk klasifikasi.

4. Melihat struktur dataset

```
print("Preview Data:")
print(df.head())
```

```
Preview Data:
         id diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0    842302         M        17.99         10.38          122.80     1001.0
1    842517         M        20.57         17.77          132.90     1326.0
2  84300903         M        19.69         21.25          130.00     1203.0
3  84348301         M        11.42         20.38           77.58      386.1
4  84358402         M        20.29         14.34          135.10     1297.0
```

```
print("\nInfo:")
print(df.info())
```

```
Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   id                       569 non-null    int64
 1   diagnosis                569 non-null    object
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness mean         569 non-null    float64
```

Penjelasan

- df.head() menampilkan 5 baris pertama untuk melihat contoh data.
- df.info() menampilkan jumlah kolom, tipe data, dan apakah ada nilai kosong.
- Langkah ini penting untuk mengetahui apakah perlu preprocessing.

5. Menentukan Kolom Target

```
target = 'diagnosis'
```

Penjelasan

- Kolom **diagnosis** adalah label yang ingin diprediksi.
- Biasanya berisi nilai:
    - M = Malignant (ganas)
    - B = Benign (jinak)

6. Label Encoding

```
if df[target].dtype == object:
    le = LabelEncoder()
    df[target] = le.fit_transform(df[target])
    print("\nLabel Encoding:", le.classes_)
```

```
...
    Label Encoding: ['B' 'M']
```

Penjelasan

- Diagnosis berupa huruf (M/B) tidak bisa diproses oleh model ML.
- LabelEncoder mengubah:
    - **B → 0**
    - **M → 1**

7. Memisahkan fitur (X) dan label (y)

```
X = df.drop(columns=[target])
y = df[target]
```

Penjelasan

- **X** adalah fitur (seluruh kolom kecuali diagnosis).
- **y** adalah kolom target yang akan diprediksi model.

8. Mengambil hanya kolom numerik

```
X = X.select_dtypes(include=[np.number])
```

Penjelasan

- Dataset Kaggle kadang memiliki kolom teks seperti ID.
- Model Naive Bayes hanya menerima data numerik, jadi kolom non-numerik dihapus otomatis.

9. Menangani nilai kosong (NaN)

```
imputer = SimpleImputer(strategy='mean')
X = imputer.fit_transform(X)
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/impute/_base.py:6
  warnings.warn(
```

Penjelasan

- Dataset kanker biasanya memiliki nilai kosong di beberapa kolom.
- Naive Bayes tidak menerima nilai NaN → harus diisi.
- Menggunakan **mean** karena dataset ini numerik.

10. Membagi data menjadi train dan test

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
```

Penjelasan

- Membagi data menjadi:
  - **80%** data training
  - **20%** data testing

- stratify=y memastikan proporsi kelas (M/B) tetap seimbang pada train dan test.

11. Melatih Model Naive Bayes

```
model = GaussianNB()
model.fit(X_train, y_train)
```

```
GaussianNB
GaussianNB()
```

Penjelasan

- GaussianNB digunakan karena fitur dataset cancer bersifat **continuous**.
- fit() melatih model berdasarkan data training.

12. Melakukan Prediksi

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

Penjelasan

- Menggunakan model yang telah dilatih untuk memprediksi diagnosis pada data testing.

13. Evaluasi Akurasi dan Laporan

```
print("\nAkurasi :", accuracy_score(y_test, y_pred))
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

```
Akurasi : 0.6228070175438597

Classification Report:
              precision    recall  f1-score   support

           0       0.63      0.99      0.77        72
           1       0.00      0.00      0.00        42

    accuracy                           0.62       114
   macro avg       0.31      0.49      0.38       114
weighted avg       0.40      0.62      0.48       114
```

Penjelasan

- **accuracy_score** → persentase prediksi benar.
- **classification_report** → menampilkan:
    - precision
    - recall
    - f1-score
    - support

Ini memberikan gambaran kualitas model.

14. Plot Confusion Matrix

```
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6,5))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title("Confusion Matrix - Gaussian Naive Bayes")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```