

Laporan Tugas Praktikum 9

Nama : Muhamad Aditia
Nim : 0110224213
Rombel : TI02
Link Github : <https://github.com/muhammadaditia433/Machine-Leraning>

Abstrak

Laporan Praktikum 9 ini membahas penerapan algoritma *Gaussian Naive Bayes* untuk klasifikasi menggunakan Python di Google Colab. Proses dimulai dari pengimporan pustaka, *mount* Google Drive, membaca dataset, dan menganalisis struktur serta kualitas data menggunakan fungsi **info()**, **describe()**, dan pengecekan nilai hilang. Data kemudian diproses melalui *label encoding* dan dibagi menjadi data latih dan uji. Model *Gaussian Naive Bayes* dilatih, diuji, serta dievaluasi menggunakan **accuracy score**, **confusion matrix**, dan **classification report**, disertai validasi silang untuk mengukur konsistensi performa. Visualisasi tambahan dibuat menggunakan Seaborn dan Matplotlib. Hasil menunjukkan bahwa model bekerja cukup baik dalam melakukan klasifikasi pada dataset yang digunakan.

1. Import Library

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

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.model_selection import cross_val_score
```

Penjelasan : Kode ini menampilkan langkah-langkah awal untuk proyek pembelajaran mesin, fokus pada pengimporan pustaka standar yang digunakan untuk analisis data dan pembangunan model Naive Bayes Gaussian.

2. Mount Ke Drive

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

Penjelasan : Kode di atas untuk menghubungkan (mount) Google Drive pribadi ke lingkungan Google Colab sehingga notebook dapat mengakses file yang tersimpan di Drive. Setelah koneksi ini berhasil diotorisasi, langkah selanjutnya yang disebut

"Membaca File Dataset" dapat dilakukan dengan menggunakan pustaka Python (seperti Pandas) untuk memuat data dari jalur Drive yang terhubung (misalnya, /content/gdrive/MyDrive/nama_file.csv) ke dalam notebook untuk analisis atau pemrosesan lebih lanjut.

3. Membaca File Dataset

```
data = pd.read_csv("/content/gdrive/MyDrive/praktikum_ml/Pertemuan-9/Data/data.csv")  
data.head()
```

Penjelasan : memuat file dataset CSV dari Google Drive ke dalam lingkungan Google Colaboratory menggunakan pustaka Python Pandas. Fungsi `pd.read_csv()` digunakan untuk membaca data dari jalur file Drive yang terhubung (seperti /content/gdrive/MyDrive/praktikum_ml/Pertemuan-9/Data/data.csv), memuatnya ke dalam DataFrame, yang kemudian dapat diprinting menggunakan metode `df.head()`. Proses ini memungkinkan analisis atau pemrosesan data lebih lanjut di dalam notebook.

4. Data Info

```
data.info()
```

Penjelasan :

Fungsi `data.info()` di pustaka Pandas Python digunakan untuk menampilkan ringkasan padat dan komprehensif dari sebuah DataFrame, membantu analisis data untuk segera memahami struktur dataset mereka.

5. Data Describe

```
data.describe()
```

Penjelasan : `data.describe()` adalah fungsi umum dalam pustaka Pandas Python yang memberikan ringkasan statistik deskriptif dari dataset atau DataFrame Anda. Secara default, fungsi ini menganalisis kolom numerik dan menyajikan metrik penting seperti rata-rata (mean), standar deviasi, nilai minimum dan maksimum, serta kuartil (persentil 25, 50, dan 75)

6. Data Isnull and Sum

```
data.isnull().sum()
```

Penjelasan : `data.isnull().sum()` digunakan dalam pustaka Pandas Python untuk menghitung jumlah total nilai yang hilang (null/NaN) di setiap kolom dalam sebuah DataFrame atau Series. Secara internal, metode `isnull()` membuat peta boolean yang menandai lokasi data yang hilang dengan True, dan kemudian `.sum()` menjumlahkan semua entri True per kolom (karena True diperlakukan sebagai 1 dalam konteks numerik), menghasilkan ringkasan yang cepat dan efisien mengenai kelengkapan dataset Anda, yang sangat penting untuk tahap pembersihan data.

7. Data df copy

```
data = df.copy()

label = LabelEncoder()
df['diagnosis'] = label.fit_transform(df['diagnosis'])

df['diagnosis'].value_counts()
```

Penjelasan : Kode Python tersebut mengimplementasikan pengkodean label pada kolom 'diagnosis' dalam DataFrame pandas menggunakan LabelEncoder dari scikit-learn untuk mengubah nilai kategorikal menjadi angka. Setelah transformasi, kode ini menggunakan `.value_counts()` untuk menghitung dan menampilkan frekuensi kemunculan setiap diagnosis yang telah diencode secara numerik tersebut.

8. Train test split

```
X = df.drop('diagnosis', axis=1)
Y = df['diagnosis']

X_train, X_test, Y_train, Y_test = train_test_split(
    X, Y, test_size=0.2, random_state=42
)

print(X.shape, X_train.shape, X_test.shape)
```

Penjelasan : Kode Python yang ditampilkan menggunakan fungsi `train_test_split()` dari pustaka `sklearn.model_selection` untuk membagi dataset menjadi dua bagian: set pelatihan (training set) dan set pengujian (testing set).

9. Gaussian NB

```
nb = GaussianNB()
nb.fit(X_train, Y_train)

Y_pred = nb.predict(X_test)
```

Penjelasan : Kode Python tersebut mengimplementasikan model klasifikasi Gaussian Naive Bayes menggunakan pustaka scikit-learn. Prosesnya dimulai dengan menginisialisasi pengklasifikasi GaussianNB(), kemudian melatih model tersebut dengan data pelatihan (X_train, Y_train) menggunakan metode fit(). Setelah pelatihan selesai, model digunakan untuk membuat prediksi pada data pengujian (X_test) menggunakan metode predict(), dan hasilnya disimpan dalam variabel Y_pred untuk evaluasi lebih lanjut.

10. Scikit learn

```
print("Accuracy:", accuracy_score(Y_test, Y_pred))
print("\nConfusion Matrix:")
print(confusion_matrix(Y_test, Y_pred))
print("\nClassification Report:")
print(classification_report(Y_test, Y_pred))
```

Penjelasan : Scikit-learn (sklearn), yang merupakan alat standar dalam machine learning untuk Python. Fungsi-fungsi spesifik yang digunakan, yaitu accuracy_score, confusion_matrix, dan classification_report, semuanya berada dalam modul sklearn.metrics. Kode ini berfungsi untuk mengevaluasi kinerja model klasifikasi dengan membandingkan nilai aktual (Y_test) dengan nilai prediksi (Y_pred), memberikan gambaran lengkap mengenai seberapa akurat model tersebut, jenis kesalahan apa yang dibuatnya, serta metrik presisi, recall, dan F1-score untuk setiap kelas.

11. Naive Bayes

```
cv_scores = cross_val_score(nb, X, Y, cv=5, scoring='accuracy')

print("\nNaive Bayes Cross Validation Accuracy:")
print("Scores:", cv_scores)
print("Mean Accuracy:", cv_scores.mean())
print("Standard Deviation:", cv_scores.std())
```

Penjelasan : Kode Python yang ditampilkan dalam gambar melakukan validasi silang (cross-validation) pada model Naive Bayes menggunakan pustaka scikit-learn. Kode ini menghitung dan menampilkan skor akurasi rata-rata dan standar deviasi di seluruh lipatan validasi.

12. Seaboarn Matpolib

```
sns.countplot(x='diagnosis', data=df)
plt.title("Distribusi Diagnosis (0 = Benign, 1 = Malignant)")
plt.show()
```

Penjelasan : Teks dan kode yang ditampilkan menjelaskan proses analisis data dan pembelajaran mesin menggunakan Python, di mana bagian teks menguraikan penggunaan pustaka scikit-learn untuk mengevaluasi model Naive Bayes dengan menghitung akurasi rata-rata dan standar deviasi melalui validasi silang. Sementara itu, blok kode Python menggunakan pustaka seaborn (`sns.countplot`) dan matplotlib (`plt.title`, `plt.show`)

Kesimpulan

Berdasarkan hasil praktikum, algoritma *Gaussian Naive Bayes* mampu melakukan klasifikasi dengan akurasi yang cukup tinggi dan konsisten. Tahapan pra-pemrosesan data seperti pemeriksaan nilai hilang, *encoding*, dan pembagian data sangat berpengaruh pada kualitas model. Evaluasi menunjukkan bahwa Naive Bayes merupakan metode yang efektif, sederhana, dan cepat untuk digunakan pada dataset seperti yang diuji dalam praktikum ini. Model ini dapat dijadikan pilihan awal dalam tugas klasifikasi sebelum menggunakan algoritma yang lebih kompleks.