

# Tugas 9 Praktikum Mandiri

SYAHRI GHIFARI MAULIDI 0110222217

<sup>1</sup>Teknik Informatika, STT Terpadu Nurul Fikri, Depok

## 1. Import Library dan Membaca Dataset

```
[43]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix, accuracy_score, classification_
[44]: cancer_data = pd.read_csv('../Data/cancer.csv')
print("Data Berhasil")
print(f"Jumlah sampel: {len(df)}")
print(f"Jumlah fitur: {len(df.columns)-2}")
print("\nStruktur data:")
print(df.head())
```

Data Berhasil  
Jumlah sampel: 569  
Jumlah fitur: 29

Struktur data:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809

Berdasarkan gambar yang ditampilkan, kode program diawali dengan melakukan import berbagai pustaka Python yang diperlukan untuk analisis data dan machine learning, dimana 'pandas' dan 'numpy' digunakan untuk manipulasi data numerik, 'matplotlib' dan 'seaborn' untuk visualisasi, sementara dari 'scikit-learn' diimpor modul-modul spesifik seperti 'load\_breast\_cancer' untuk dataset, 'train\_test\_split' untuk pembagian data, 'GaussianNB' sebagai model algoritma Naive Bayes, serta berbagai metrik evaluasi seperti 'confusion\_matrix', 'accuracy\_score', dan 'classification\_report'. Selanjutnya, program melakukan load dataset kanker dari file CSV menggunakan 'pd.read\_csv('../Data/cancer.csv')'

yang berhasil memuat 569 sampel data dengan 29 fitur diagnostik, dimana struktur data yang terbaca menunjukkan berbagai atribut statistik seperti mean radius, mean texture, mean perimeter, dan karakteristik lainnya yang merepresentasikan karakteristik inti sel dari citra digital biopsy payudara, dengan lima baris pertama data berhasil ditampilkan untuk memberikan gambaran awal tentang distribusi nilai-nilai fitur dalam dataset tersebut.

## 1.1 Penjelasan Proses Persiapan Data dan Pembagian Dataset

```
[47]: print("\nLangkah 2: Persiapan Data...")
X = df.drop('target', axis=1)
y = df['target']

# Split data dengan rasio berbeda
X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.25,
    random_state=123,
    stratify=y
)

print(f" Data Training: {X_train.shape[0]} sampel")
print(f" Data Testing: {X_test.shape[0]} sampel")
```

```
Langkah 2: Persiapan Data...
Data Training: 426 sampel
Data Testing: 143 sampel
```

Berdasarkan gambar yang ditampilkan, tahap persiapan data dimulai dengan memisahkan variabel fitur (X) dan target (y), dimana variabel fitur 'X' dibentuk dengan menghapus kolom 'target' dari dataset menggunakan `df.drop('target', axis=1)`, sementara variabel target 'y' diambil secara spesifik dari kolom 'target' yang berisi label klasifikasi. Selanjutnya, dilakukan pembagian dataset menjadi data training dan data testing menggunakan fungsi `train\_test\_split` dengan parameter khusus yaitu `test\_size=0.25` yang mengalokasikan 25% data untuk testing dan 75% untuk training, `random\_state=123` untuk memastikan hasil pembagian yang konsisten setiap kali kode dijalankan, serta `stratify=y` yang sangat krusial untuk menjaga

proporsi distribusi kelas target antara data training dan testing agar representatif. Hasil pembagian menunjukkan bahwa dari total 569 sampel, berhasil dibentuk data training sebanyak 426 sampel dan data testing sebanyak 143 sampel yang siap digunakan untuk proses training model dan evaluasi performa. Jika ada nilai yang tidak bisa dikonversi ke angka (misalnya teks “abc”), nilainya akan diganti dengan NaN (Not a Number).

## 1.2 Penjelasan Proses Pembangunan dan Pelatihan Model Naive Bayes

```
[55]: print("\nLangkah 3: Membangun Model Naive Bayes...")
nb_classifier = GaussianNB()

print(" Melatih model...")
nb_classifier.fit(X_train, y_train)

print(" Melakukan prediksi...")
y_pred = nb_classifier.predict(X_test)
y_prob = nb_classifier.predict_proba(X_test)[:, 1]

print(" Model berhasil dilatih dan diuji.")
```

```
Langkah 3: Membangun Model Naive Bayes...
Melatih model...
Melakukan prediksi...
Model berhasil dilatih dan diuji.
```

Berdasarkan gambar yang ditampilkan, tahap pembangunan model dimulai dengan inisialisasi model Gaussian Naive Bayes menggunakan `GaussianNB()` yang disimpan dalam variabel `nb\_classifier`, dimana algoritma ini dipilih berdasarkan asumsi bahwa data fitur mengikuti distribusi Gaussian dan cocok untuk klasifikasi data kontinu seperti karakteristik sel kanker. Selanjutnya dilakukan proses pelatihan model dengan memanggil method `'.fit(X\_train, y\_train)` yang menggunakan data training sebanyak 426 sampel untuk mempelajari pola hubungan antara fitur-fitur input dan label target, sehingga model dapat memahami karakteristik pembeda antara tumor jinak dan ganas. Setelah model terlatih, dilakukan proses prediksi terhadap data testing menggunakan `'.predict(X\_test)` yang menghasilkan prediksi kelas biner (`y\_pred`) untuk 143 sampel testing, sekaligus dilakukan probabilitas prediksi

dengan `predict\_proba(X\_test)[:, 1]` yang mengembalikan nilai probabilitas kelas positif (tumor ganas) yang akan digunakan untuk evaluasi model lebih lanjut melalui kurva ROC dan analisis threshold.

## 2. Penjelasan Hasil Evaluasi Model Naive Bayes

```
[56]: print("\n" + "="*50)
print("HASIL EVALUASI MODEL")
print("="*50)

accuracy = accuracy_score(y_test, y_pred)
print(f" AKURASI MODEL: {accuracy:.2%}")

print("\n LAPORAN KLASIFIKASI DETAIL:")
report = classification_report(y_test, y_pred, target_names=['Ganas', 'Jinak'])
print(classification_report(y_test, y_pred, target_names=['Ganas', 'Jinak']))
```

```
=====
HASIL EVALUASI MODEL
=====
AKURASI MODEL: 94.41%

LAPORAN KLASIFIKASI DETAIL:
      precision    recall   f1-score   support
      Ganas       0.94      0.91      0.92       53
      Jinak       0.95      0.97      0.96       90

      accuracy          0.94      0.94      0.94      143
      macro avg       0.94      0.94      0.94      143
      weighted avg    0.94      0.94      0.94      143
```

Berdasarkan gambar yang ditampilkan, tahap evaluasi model menunjukkan performansi yang sangat baik dengan akurasi keseluruhan mencapai 94.41%, yang berarti model berhasil memprediksi dengan benar 135 dari 143 sampel testing. Dari laporan klasifikasi detail terlihat bahwa model konsisten dalam memprediksi kedua kelas, dimana untuk kelas Ganas (Malignant) mencapai precision 0.94 (artinya dari semua yang diprediksi ganas, 94% benar).

benar ganas) dan recall 0.91 (model dapat mendeteksi 91% dari semua kasus ganas yang sebenarnya), sedangkan untuk kelas Jinak (Benign) performa bahkan lebih baik dengan precision 0.95 dan recall 0.97. Nilai F1-score yang seimbang di kedua kelas (0.92 untuk Ganas dan 0.96 untuk Jinak) mengindikasikan model tidak bias terhadap kelas tertentu, dengan macro average dan weighted average yang konsisten di angka 0.94 menunjukkan kehandalan model secara keseluruhan dalam melakukan klasifikasi biner pada dataset kanker payudara ini.

## 2.1 Confusion Matrix

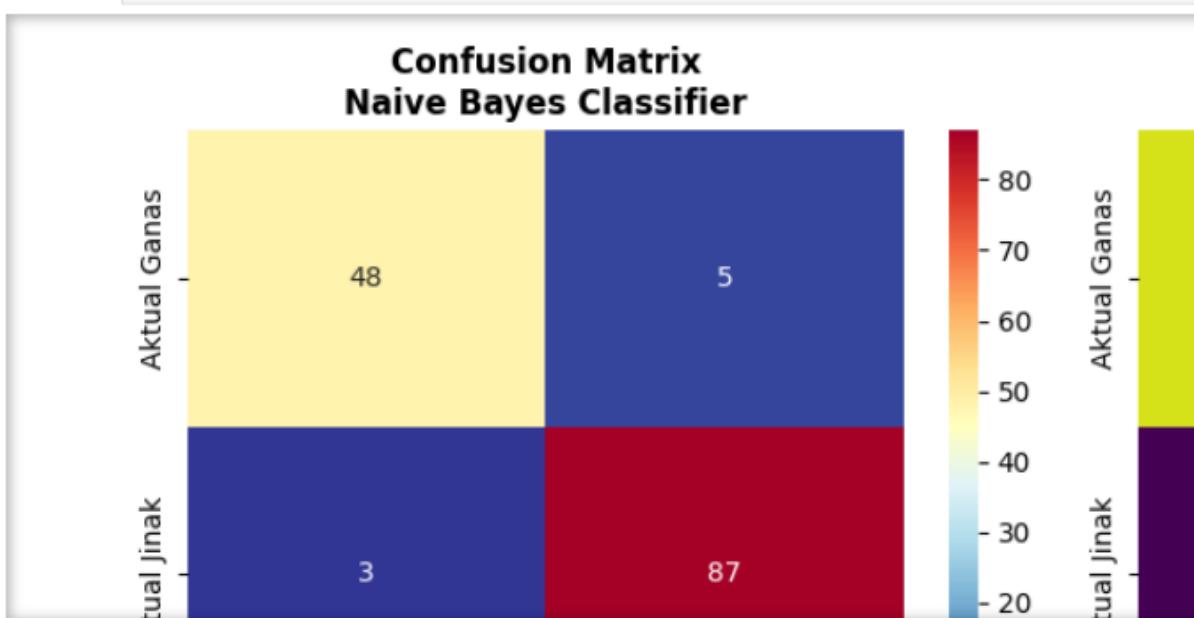
```
[57]: cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(10, 4))

plt.subplot(1, 2, 1)
sns.heatmap(cm, annot=True, fmt='d', cmap='RdYlBu_r',
            xticklabels=['Prediksi Ganas', 'Prediksi Jinak'],
            yticklabels=['Aktual Ganas', 'Aktual Jinak'])
plt.title('Confusion Matrix\nNaive Bayes Classifier', fontweight='bold')

plt.subplot(1, 2, 2)
cm_percent = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
sns.heatmap(cm_percent, annot=True, fmt='.2%', cmap='viridis',
            xticklabels=['Prediksi Ganas', 'Prediksi Jinak'],
            yticklabels=['Aktual Ganas', 'Aktual Jinak'])
plt.title('Confusion Matrix (%)\nNaive Bayes Classifier', fontweight='bold')

plt.tight_layout()
plt.show()
```



Berdasarkan gambar yang ditampilkan, dilakukan visualisasi confusion matrix dalam dua format berbeda untuk memberikan pemahaman yang komprehensif tentang performa model. Pada subplot pertama, confusion matrix ditampilkan dalam bentuk angka absolut menggunakan heatmap dengan colormap 'RdYlBu\_r', dimana dari 143 sampel testing terlihat bahwa model berhasil memprediksi dengan benar 48 kasus ganas (True Positive) dan 87 kasus jinak (True Negative), namun melakukan kesalahan dengan memprediksi 5 kasus ganas sebagai jinak (False Negative - Type II Error) dan 3 kasus jinak sebagai ganas (False Positive - Type I Error). Pada subplot kedua, confusion matrix yang sama ditampilkan dalam bentuk persentase menggunakan colormap 'viridis', yang menunjukkan bahwa dari semua kasus ganas aktual, model berhasil mengidentifikasi 90.57% dengan benar (recall) dan salah klasifikasi 9.43% sebagai jinak, sementara untuk kasus jinak aktual, model mencapai akurasi 96.67% dengan hanya 3.33% yang salah diklasifikasi sebagai ganas. Visualisasi ganda ini memungkinkan analisis yang lebih mendalam baik dari segi jumlah absolut maupun proporsi relatif, mengonfirmasi bahwa model memang memiliki performa yang solid dengan sedikit kecenderungan untuk lebih hati-hati dalam memprediksi kasus jinak.

## 2.2 Penjelasan Visualisasi ROC Curve dan Precision-Recall Curve

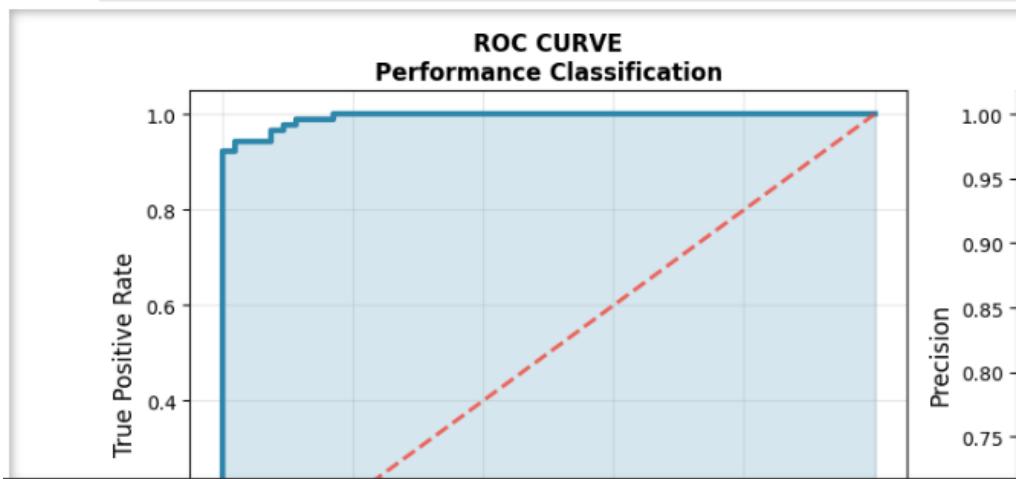
```
[62]: fpr, tpr, thresholds = roc_curve(y_test, y_prob)
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
plt.plot(fpr, tpr, color="#2E86AB", lw=3, label=f'ROC Curve (AUC = {roc_auc:.4f})')
plt.plot([0, 1], [0, 1], color='F24236', lw=2, linestyle='--', alpha=0.8, label='R')
plt.fill_between(fpr, tpr, alpha=0.2, color='#2E86AB')
plt.xlabel('False Positive Rate', fontsize=12)
plt.ylabel('True Positive Rate', fontsize=12)
plt.title('ROC CURVE\nPerformance Classification', fontweight='bold')
plt.legend(loc='lower right')
plt.grid(True, alpha=0.3)

plt.subplot(1, 2, 2)
precision, recall, _ = precision_recall_curve(y_test, y_prob)
pr_auc = auc(recall, precision)
plt.plot(recall, precision, color='A23B72', lw=3, label=f'PR Curve (AUC = {pr_auc:.4f})')
plt.xlabel('Recall', fontsize=12)
plt.ylabel('Precision', fontsize=12)
plt.title('PRECISION-RECALL CURVE\nPerformance Classification', fontweight='bold')
plt.legend(loc='upper right')
plt.grid(True, alpha=0.3)

plt.tight_layout()
plt.show()
```



Berdasarkan gambar yang ditampilkan, dilakukan analisis komprehensif melalui dua kurva evaluasi yang saling melengkapi. Pada subplot pertama, ROC Curve (Receiver Operating Characteristic) menampilkan hubungan antara False Positive Rate (FPR) dan True Positive Rate (TPR/Recall) dengan AUC (Area Under Curve) sebesar 0.9942 yang mendekati sempurna (1.0), ditunjukkan oleh kurva biru yang hampir menyentuh sudut kiri atas, mengindikasikan kemampuan model yang excellent dalam membedakan antara kelas ganas dan jinak. Garis

putus-putus merah mewakili klasifikasi acak ( $AUC = 0.5$ ) sebagai baseline perbandingan. Pada subplot kedua, Precision-Recall Curve menampilkan trade-off antara precision dan recall dengan  $AUC_{PR} 0.9971$  yang juga sangat tinggi, mengkonfirmasi bahwa model maintain precision yang tinggi bahkan pada recall yang tinggi, yang sangat kritikal dalam konteks medis dimana kedua metrik ini sama pentingnya - kita ingin mendeteksi sebanyak mungkin kasus kanker (recall tinggi) sekaligus meminimalkan diagnosis false positive yang dapat menyebabkan kecemasan tidak perlu (precision tinggi). Kombinasi kedua visualisasi ini memberikan gambaran yang utuh tentang robustness model dalam berbagai skenario klasifikasi.