

Lembar Kerja Praktikum Pertemuan 7
Evaluasi Klasifikasi

Anggota Kelompok 4:

- **Ismy Fana Fillah (G6401211001)**
- **Tita Madriyanti (G6401211120)**
- **Halida Fiadnin (G6401211142)**
- **Viragita Athaya Haura (G6401211116)**

A. Evaluasi Klasifikasi

Melakukan evaluasi terhadap hasil evaluasi penting untuk dilakukan untuk menguji apakah suatu metode klasifikasi memiliki performance yang baik atau tidak. Oleh karena itu, diperlukan metode/teknik untuk melakukan evaluasi tersebut. Terdapat beberapa cara metrik untuk mengukur performa proses klasifikasi. Beberapa metrik tersebut antara lain :

- a. Akurasi (accuracy)
- b. Sensitivitas (sensitivity)
- c. Spesifisitas (specificity)
- d. Recall
- e. Precision
- f. F1-Score
- g. ROC AUC Curve

Untuk mendapatkan nilai-nilai tersebut, diperoleh dengan terlebih dahulu membuat matrik konfusi atau confusion matrix. Sebagai ilustrasi, diberikan matrik konfusi sebagai berikut :

	Aktual	
	TP	FP
	FN	TN

Formula untuk menentukan beberapa metrik adalah sebagai berikut :

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$$

$$Sensitivity = Recall = \frac{TP}{TP+FN}$$

$$Specificity = \frac{TN}{TN+FP}$$

$$Precision = \frac{TP}{TP+FP}$$

$$F1 - Score = 2 \times \frac{recall \cdot precision}{recall+precision}$$

ROC dan AUC menyajikan performa klasifikasi dalam bentuk grafik / kurva. Semakin besar Area Under Curve, maka performa klasifikasi semakin baik.

B. Penggunaan Library Python untuk Evaluasi Klasifikasi

Untuk melakukan evaluasi, Python memiliki Library yang dapat digunakan yakni Scikit learn. Silahkan perhatikan dan mencoba source code script python di bawah ini :

```
# Step 1: Load the dataset
from sklearn import datasets

data = datasets.load_breast_cancer()
X = data.data
y = data.target
target_names = data.target_names
print('X:',X[0],'\ny:',y[:5],'\ntarget_names:',target_names)

import pandas as pd

df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = data.target

df['target_names'] = data.target_names[df['target']]

#print(df.head())

from sklearn.model_selection import train_test_split

# Step 2: Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

# Step 3: Perform hyperparameter tuning with GridSearchCV
```

```

# Instantiate the model
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier()

# Tune hyper parameters with GridSearchCV
param_grid = {"n_neighbors": [3, 5, 7, 9]}

from sklearn.model_selection import GridSearchCV
grid_search = GridSearchCV(knn, param_grid, cv=5)

grid_search.fit(X_train, y_train)
best_k = grid_search.best_params_["n_neighbors"]

# Step 4: Train the model
knn = KNeighborsClassifier(n_neighbors=best_k)
knn.fit(X_train, y_train)

# Step 5: Make predictions
y_pred = knn.predict(X_test)

# Evaluate the performance
from sklearn.metrics import (
    accuracy_score,
    precision_score,
    recall_score,
    f1_score,
    roc_auc_score
)

# Step 6: Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_pred)

```

```

print(f'Accuracy: {accuracy}')
print(f'Sensitivity / Recall: {recall}')
print(f'Precision: {precision}')
print(f'F1-Score: {f1}')
print(f'ROC AUC: {roc_auc}')

from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt

# Step 6: Plot the confusion matrix
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=target_names)
disp.plot()
disp.ax_.set(title='Confusion Matrix')
plt.show()

import matplotlib.pyplot as plt
from sklearn.metrics import RocCurveDisplay

# Step 8: Plot the ROC AUC curve
y_scores = knn.predict_proba(X_test)
roc_display = RocCurveDisplay.from_estimator(knn, X_test, y_test)
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC AUC Curve')
plt.show()

```

Makna dan insight dari masing-masing metrik klasifikasi tersebut:

Akurasi (Accuracy):

- Makna: Akurasi mengukur seberapa sering model klasifikasi membuat prediksi yang benar secara keseluruhan dari semua kelas.
- Insight: Akurasi memberikan gambaran tentang seberapa baik model secara keseluruhan dalam memprediksi kelas-kelas yang benar. Namun, dalam kasus di mana kelas-kelas

memiliki distribusi yang tidak seimbang, akurasi mungkin tidak mencerminkan performa model secara menyeluruh.

Sensitivitas (Sensitivity):

- Makna: Sensitivitas (atau juga dikenal sebagai True Positive Rate atau Recall) mengukur seberapa baik model dapat mengidentifikasi kasus positif sejati dari semua kasus yang benar positif.
- Insight: Sensitivitas penting ketika mengutamakan pengurangan false negative, misalnya dalam mendeteksi penyakit atau anomali. Sensitivitas yang tinggi berarti model mampu mengidentifikasi sebagian besar kasus positif.

Spesifisitas (Specificity):

- Makna: Spesifisitas (atau True Negative Rate) mengukur seberapa baik model dapat mengidentifikasi kasus negatif sejati dari semua kasus yang sebenarnya negatif.
- Insight: Spesifisitas penting ketika fokus pada mengurangi false positive. Contohnya adalah dalam skrining untuk mengkonfirmasi kesehatan, di mana mungkin lebih baik untuk meremehkan beberapa kasus positif palsu daripada melewatkan kasus negatif.

Recall:

- Makna: Recall, yang juga dikenal sebagai True Positive Rate atau Sensitivitas, adalah proporsi kasus positif yang diidentifikasi secara benar dari semua kasus yang sebenarnya positif.
- Insight: Recall membantu memahami kemampuan model untuk menangkap semua kasus positif. Nilai recall yang tinggi menunjukkan bahwa model cenderung tidak melewatkan banyak kasus positif, meskipun mungkin ada peningkatan false positive.

Precision:

- Makna: Precision adalah proporsi kasus positif yang diidentifikasi secara benar dari semua kasus yang diprediksi positif oleh model.
- Insight: Precision membantu mengukur seberapa tepat model dalam memprediksi kasus positif. Precision yang tinggi berarti sedikit sekali hasil yang sebenarnya negatif yang salah diprediksi sebagai positif.

F1-Score:

- Makna: F1-Score adalah rata-rata harmonik dari precision dan recall. Ini memberikan keseimbangan antara kedua metrik tersebut.
- Insight: F1-Score berguna ketika kita ingin mencari keseimbangan antara precision dan recall. Nilai F1-Score yang tinggi menunjukkan bahwa model memiliki precision dan recall yang baik secara bersamaan, bukan hanya satu atau yang lain. F1-Score berguna ketika kelas-kelas tidak seimbang.

Tugas

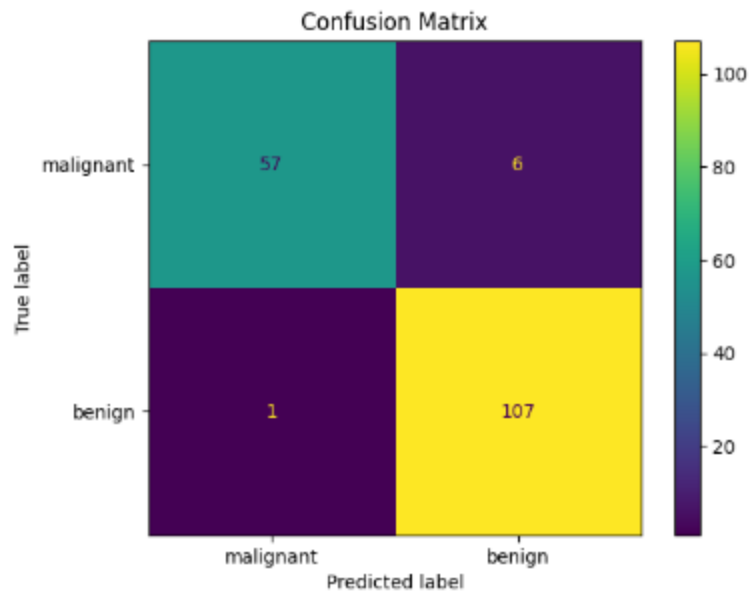
1. Ketik ulang script di atas

Link colab :

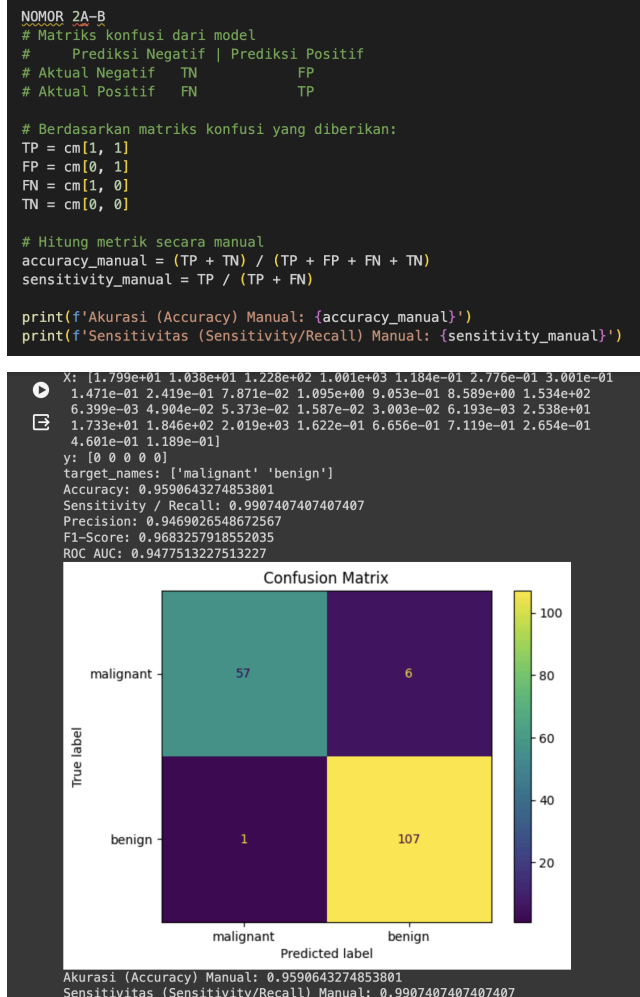
<https://colab.research.google.com/drive/1cNFjmZulE-MfqvhwVWWhFXZ6tun9k56V?usp=sharing>

Output :

Accuracy: 0.9598643274853881
Sensitivity / Recall: 0.9987487487487487
Precision: 0.9469026548672567
F1-Score: 0.9683257918552035
ROC AUC: 0.9477513227513227



2. Lakukan perhitungan secara manual berdasarkan matrik konfusi yang diperoleh untuk menghitung metrik
 - a. Akurasi (accuracy)
 - b. Sensitivitas (sensitivity)



c. Spesifisitas (specificity)

$$Specificity = \frac{TN}{TN+FP} = \frac{107}{107+1} = \frac{107}{108} = 0.990740741$$

d. Recall

$$Recall = \frac{TP}{TP+FN} = \frac{57}{57+6} = \frac{57}{63} = 0.904761905$$

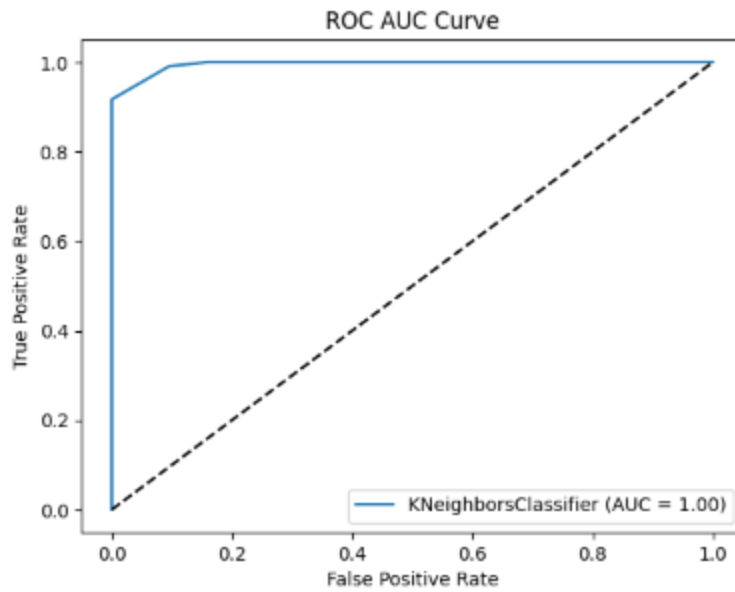
e. Precision

$$Precision = \frac{TP}{(TP++FP)} = \frac{107}{(107+6)} = 94.69026549\%$$

f. F1-Score

$$F1 - Score = \frac{2 \times Precision \times Recall}{(Precision + Recall)} = \frac{2 \times 94.69026549\% \times 99.07407407407407\%}{(94.69026549\% + 99.07407407407407\%)} = 96.83257919\%$$

3. Perhatikan kurva ROC AUC yang dihasilkan.



4. Jelaskan apa makna yang tersirat dari hasil klasifikasi yang didapatkan pada soal sebelumnya berdasarkan keluaran setiap metrik.