

**PENGUKURAN PERFORMA MODEL MASALAH KLASIFIKASI MENGGUNAKAN
METODE CONFUSION MATRIKS**

*Makalah Ini Disusun untuk Memenuhi Tugas Mata Kuliah
Metode Kecerdasan Buatan*

Dosen Pengampu : Muhammad Irvan Septiar Musti, S.Si, M.Si



Di susun oleh :

Fida Suci Rahmani	11190940000027
Rosa Amalia Nursinta	11190940000041
Elviana Saputri	11190940000043
Ghina Rahmah	11190940000053
Meissy Astariva Putri	11190940000063

**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH JAKARTA
2022**

KATA PENGANTAR

Assalammu'alaikum Warahmatullahi Wabarakatuh

Puji syukur mari kita panjatkan kepada Allah SWT yang Maha Esa karena atas berkat rahmat dan karunia-Nya kami dapat menyelesaikan penyusunan makalah dengan judul "Pengukuran Performa Model Masalah Klasifikasi Menggunakan Metode Confusion Matriks" dapat dikerjakan dengan tepat waktu dan dengan sebaik-baiknya.

Makalah ini merupakan tugas yang harus diselesaikan dalam mata kuliah Metode Kecerdasan Buatan dengan dosen pengampunya adalah Bapak Muhammad Irvan Septiar Musti, S.Si, M.Si. Harapan kami tentunya berharap agar makalah ini dapat membantu dan menambah wawasan bagi para pembaca.

Kami menyadari penuh bahwa dalam penyusunan project ini masih terdapat banyak kekurangan. Kami mengharapkan kritik dan saran yang membangun dari pembaca agar kedepannya kami bisa melakukan perbaikan untuk mendapatkan hasil yang lebih baik. Terakhir, kami berharap semoga laporan ini dapat bermanfaat dan dapat tercapai sesuai dengan yang diharapkan.

Wassalammu'alaikum Warahmatullahi Wabarakatuh

Tangerang Selatan, 19 April 2022

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring berjalannya waktu, perkembangan teknologi dan data semakin berkembang pesat. Perkembangan data yang semakin tinggi mendorong untuk memanfaatkan data. Dalam penggalian informasi maupun pengetahuan. Salah satu pendekatan yang dapat digunakan untuk menganalisis sekumpulan data adalah klasifikasi. Klasifikasi merupakan salah satu teknik data mining yang digunakan untuk membangun suatu model dari sampel data yang belum terklasifikasi untuk digunakan mengklasifikasi sampel data baru ke dalam kelas-kelas yang sejenis [1]. Klasifikasi termasuk ke dalam supervised learning karena menggunakan sekumpulan data untuk dianalisis terlebih dahulu, kemudian pola dari hasil analisis tersebut digunakan untuk pengklasifikasian data uji. Proses klasifikasi data terdiri dari pembelajaran dan Klasifikasi. Pada pembelajaran data training dianalisis menggunakan algoritma klasifikasi, selanjutnya pada klasifikasi digunakan data testing untuk memastikan tingkat akurasi dari rule Klasifikasi yang digunakan. Teknik klasifikasi dibagi menjadi lima kategori berdasarkan perbedaan konsep matematika, yaitu berbasis statistik, berbasis jarak, berbasis pohon keputusan, berbasis jaringan syaraf, dan berbasis rule [2]. Klasifikasi termasuk ke dalam supervised learning karena menggunakan sekumpulan data untuk dianalisis terlebih dahulu, kemudian pola dari hasil analisis tersebut digunakan untuk pengklasifikasian data uji. Proses klasifikasi data terdiri dari pembelajaran dan klasifikasi. Pada pembelajaran data training dianalisis menggunakan algoritma klasifikasi, selanjutnya pada klasifikasi digunakan data testing untuk memastikan tingkat akurasi dari rule klasifikasi yang digunakan.

Supervised learning merupakan pendekatan machine learning yang ditentukan berdasarkan penggunaan dataset berlabel (labeled dataset). Dalam dataset ini, terdapat sebuah "label", yaitu satu kolom yang menjadi target output model. Dalam supervised learning, model dilatih menggunakan dataset training dan diawasi (supervise) untuk melakukan klasifikasi atau prediksi sesuai dengan output berupa data berlabel yang sudah ditentukan sebelumnya, berdasarkan pola yang ada dalam data training [3]. Sistem dalam klasifikasi diharapkan mampu melakukan klasifikasi semua Set data dengan benar, namun tidak dapat dipungkiri bahwa kesalahan akan terjadi. Dalam proses pengklasifikasian tersebut sehingga perlunya dilakukan pengukuran Kinerja dari sistem klasifikasi tersebut [4].

1.2 Rumusan Masalah

1. Bagaimana implementasi supervise Learning Classification terhadap penelitian pengklasifikasian bunga iris?
2. Berapa akurasi pengenalan jenis bunga iris menggunakan metode confusion matrix dalam mengenali jenisnya?

1.3 Tujuan Makalah

1. Mengetahui implementasi supervise Learning Classification terhadap penelitian pengklasifikasian bunga iris
2. Dapat menghitung ketepatan akurasi confusion matrix pada data test untuk memprediksi suatu objek.

BAB II

TINJAUAN PUSTAKA

2.1. Supervised Learning – Classification

Machine Learning merupakan salah satu cabang dari disiplin ilmu kecerdasan buatan (Artificial Intelligence) yang membahas bagaimana sistem dibangun berdasarkan pada data. Jadi machine learning merupakan proses komputer untuk belajar dari data (learn from data). Jika tidak ada data, komputer tidak akan bisa belajar. Salah satu teknik aplikasi pada machine learning adalah supervised learning. Klasifikasi merupakan supervised learning, yang merupakan model prediksi dimana hasil prediksinya bersifat diskrit. Pengukuran performa dari suatu model klasifikasi dapat dilakukan dengan membandingkan nilai aktual dan nilai prediksi. Maka dalam mengukur suatu performa model dibutuhkan alat dalam pengukuran model tersebut yaitu dengan menggunakan confusion matrix.[5]

2.2. Confusion Matrix

Confusion Matrix adalah pengukuran performa untuk masalah klasifikasi machine learning dimana keluaran dapat berupa dua kelas atau lebih. Confusion Matrix adalah tabel dengan 4 kombinasi berbeda dari nilai prediksi dan nilai aktual. Ada empat istilah yang merupakan representasi hasil proses klasifikasi pada confusion matrix yaitu *True Positif*, *True Negatif*, *False Positif*, dan *False Negatif*.[6]

Contoh Confusion Matrix :

Suatu perusahaan membuat sebuah model yang dilatih untuk memprediksi apakah seorang karyawan di perusahaan tsb terkena covid-19 atau tidak. Dengan asumsi perusahaan tsb mempunyai 175 karyawan. Dari model classifier memprediksi karyawan positif covid-19 sebanyak 145 dan karyawan negatif covid-19 sebanyak 30 tetapi pada kenyataannya, karyawan positif covid-19 sebanyak 150 dan karyawan negatif covid-19 sebanyak 25 (dapat dilihat pada Tabel 1).

Tabel 1 Contoh Confusion Matrix

n= 175	Aktual : Positif (1)	Aktual : Negatif (0)
Prediksi : Positif (1)	TP: 125	FP: 20
Prediksi : Negatif (0)	FN: 25	TN: 5
	150	25

2.3. True Positive, True Negative, False Positive, False Negative

- True Positive (TP)
Interpretasi: Anda memprediksi positif dan itu benar. Anda memprediksikan bahwa seorang wanita hamil dan wanita tsb memang benar hamil.
- True Negative (TN)
Interpretasi: Anda memprediksi negatif dan itu benar. Anda memprediksikan bahwa seorang pria tidak hamil dan benar ya pria kan tidak mungkin hamil
- False Positive (FP) (Kesalahan Tipe 1)
Interpretasi: Anda memprediksi positif dan itu salah. Anda memprediksikan bahwa seorang pria hamil tetapi tidak mungkin pria bisa hamil
- False Negative (FN) (Kesalahan Tipe 2 Kesalahan tipe 2 ini sangat berbahaya)
Interpretasi: Anda memprediksi negatif dan itu salah. Anda memperkirakan bahwa seorang wanita tidak hamil tetapi sebenarnya wanita tsb hamil.

Dari contoh di atas dapat digambarkan bahwa:

- Nilai Prediksi adalah keluaran dari program dimana nilainya Positif dan Negatif
- Nilai Aktual adalah nilai sebenarnya dimana nilainya True dan False

Contoh TP, TN, FP, FN dari kasus covid sebelumnya :

- True Positive (TP): kita memprediksi karyawan positif covid-19 dan memang benar karyawan tsb positif covid-19
- True Negative (TN): kita memprediksi karyawan negative covid-19 dan memang benar karyawan tsb negatif covid-19.
- False Positive (FP): kita memprediksi karyawan positif covid-19 dan ternyata prediksi salah, ternyata karyawan negatif covid-19
- False Negative (FN) : kita memprediksi karyawan negatif covid-19 dan ternyata prediksi salah, ternyata karyawan tsb positif covid-19. Seperti telah dijelaskan di atas bahwa FN merupakan kesalahan tipe 2 dimana kesalahan ini sangat berbahaya. Contoh : karyawan di prediksi negatif covid-19 padahal ternyata karyawan positif covid-19 maka karyawan tersebut terlambat mengetahui keadaan sebenarnya sehingga tidak segera dilakukan tindakan pencegahan pengobatan dan isolasi mandiri. Dimana karyawan dapat menularkan virus covid-19 ke banyak karyawan lainnya dalam perusahaan dan dapat menyebabkan kematian untuk karyawan lain juga karyawan tsb.

2.4. Accuracy, Precision, Recall, Specificity

Accuracy adalah skala prediksi yang dihitung sebagai dua akurasi prediksi dibagi dengan jumlah keseluruhan data. Rumus perhitungan akurasi dapat dilihat dari persamaan di bawah ini [7].

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

Precision adalah skala prediksi yang benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif yang dapat dihitung dengan jumlah prediksi yang benar positif dibagi dengan jumlah keseluruhan prediksi positif. Rumus perhitungan precision dapat dilihat dari persamaan di bawah ini [8].

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

Recall adalah skala prediksi yang benar positif akurat dibandingkan dengan keseluruhan data yang benar positif yang dapat dihitung dengan jumlah prediksi positif yang akurat dibagi dengan jumlah keseluruhan data yang benar positif. Rumus perhitungan recall dapat dilihat dari persamaan di bawah ini [8].

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

Specificity adalah memperkirakan kemampuan suatu tes untuk menjadi negatif ketika kondisi sebenarnya tidak ada. Rumus perhitungan specificity dapat dilihat dari persamaan di bawah ini [9].

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

2.5. F beta Score

F-beta score merupakan salah satu tes keakuratan model dalam analisis statistik dengan cara menghitung rata-rata harmonik [10] dari precision and recall, biasanya tes ini digunakan untuk mengevaluasi model binary classification dan multiclass classification. Aplikasi F-score dapat diterapkan dalam mesin pencari informasi, dimana mesin ini bekerja untuk menampilkan informasi berdasarkan keyword yang dibutuhkan user, tetapi tidak semua dokumen yang mengandung keyword tersebut boleh ditampilkan, sehingga mesin ini menyaring dokumen yang paling relevan terhadap pencarian.

Nilai F-beta berada di interval nol sampai satu. Nilai terburuk yaitu 0(nol) dan jika nilainya mendekati angka satu maka semakin baik modelnya. Nilai itu didapatkan dengan mensubstitusikan nilai Precision dan Recall yang telah didapatkan dan nilai β , melalui rumus di bawah ini:

$$F - \beta = \frac{(1 + \beta^2) Precision \times Recall}{Precision + Recall}$$

- Nilai β yang sering digunakan ada tiga, yaitu; 0.5, 1, dan 2. Penentuan nilai β didasarkan oleh tujuan meminimalkan false negative atau false positive.
- F-0.5 ($\beta = 0.5$), untuk penggunaan beta bernilai 0.5 berarti hasil yang diinginkan lebih fokus pada mengurangi false positive atau bobot *precision* lebih besar dibanding *recall*.

$$F - 0.5 = \frac{1.25 * Precision * Recall}{Precision + Recall}$$

- F-1 ($\beta = 1$), penggunaan $\beta = 1$ merupakan default dimana hasilnya berbobot sama antara *Precision* dan *Recall*.

$$F - 1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

- F-2 ($\beta = 2$), untuk penggunaan beta bernilai 2 berarti hasil yang diinginkan lebih fokus pada mengurangi false negative atau bobot *recall* lebih besar dibanding *precision*.

$$F - 2 = \frac{5 * Precision * Recall}{Precision + Recall}$$

Python sebagai salah satu bahasa pemrograman juga menyediakan *library* yang dapat memudahkan untuk menghitung F-beta score.

Model multiclass classification juga dapat dievaluasi F-score dengan formula yang berbeda. Ada tiga cara untuk melihat nilai f-beta multiclass nilai akhir dengan rata-rata mikro, makro, atau default average.[11]

➤ *Macro average*

Menghitung F-1 score berdasarkan hitungan perbaris 'kenyataan'(yang bukan dugaan).

$$macro F - 1 = \frac{2 * macro Precision * macro Recall}{macro Precision + macro Recall}$$

$$macro Precision = \frac{P_0 + P_1 + \dots + P_n}{n}, n = 0, 1, 2, \dots, n$$

$$P_n = \frac{TP_n}{TP_n + FP_n}$$

$$macro Recall = \frac{R_0 + R_1 + \dots + R_n}{n}, n = 0, 1, 2, \dots, n$$

$$R_n = \frac{TP_n}{TP_n + FN_n}$$

➤ *Micro average*

Micro average menghitung dengan cara yang lebih menyeluruh.

$$micro\ Precision = \frac{\sum_0^n TP_n}{\sum_0^n TP_n + \sum_0^n FP_n}$$

$$micro\ Recall = \frac{\sum_0^n TP_n}{\sum_0^n TP_n + \sum_0^n FN_n}$$

➤ *Default average*

Menghitung F-1 score dengan cara ini sebenarnya hampir mirip dengan cara mencari berdasarkan macro F-1 score yaitu perbaris 'kenyataan'.

$$F - 1 = \frac{F_0 + F_1 + \dots + F_n}{n}$$

$$F_n = \frac{2 * P_n * R_n}{P_n + R_n}$$

$$P_n = \frac{TP_n}{TP_n + FP_n}$$

$$R_n = \frac{TP_n}{TP_n + FN_n}$$

Untuk data yang muticlass dan balance (banyak data sama) hasil dari ketiga cara diatas tidak akan berbeda jauh, tetapi untuk data yang muticlass dan imbalance disarankan agar menggunakan cara micro average karena biasanya hasilnya yang paling baik.

2.6. AUC & ROC

Pada praktiknya, receiver operator characteristic (ROC) curve digunakan untuk menunjukkan perbandingan predictor dengan keluaran yang sebenarnya. ROC curve merupakan alat untuk menganalisa dengan menggunakan hubungan antara sensitivity dan specificity dari klasifikasi binari, disebut dengan ROC analysis. Sensivity atau true positive rate mengukur proporsi nilai positif yang diklasifikasikan dengan benar, sedangkan Specificity atau false positive rate mengukur proporsi negative yang diklasifikasikan negative. Garis horizontal menggambarkan nilai false positif dan garis vertikal menggambarkan nilai true positive (Vercellis, 2006). Kurva ROC dibagi dalam dua dimensi, dimana true positive rate di plot pada sumbu Y dan false positive rate

di plot pada sumbu X. Biasanya, analisis ROC menunjukkan bagaimana sensitivity berubah dengan spesifisitas yang bervariasi untuk ambang berbeda, serta mempertimbangkan positif palsu dan negatif palsu secara seimbang (Flach, 2016).

Dalam analisis ROC, menentukan klasifikasi yang lebih baik digunakan metode menghitung area under the curve (AUC) dimana diartikan sebagai probabilitas. AUC mengukur kinerja diskriminatif dengan memperkirakan probabilitas output dari sampel yang dipilih secara acak dari populasi positif atau negatif, semakin besar AUC, semakin kuat klasifikasi yang digunakan. Karena AUC adalah bagian dari daerah unit persegi, nilainya akan selalu antara 0,0 dan 1,0. AUC sering digunakan untuk mengukur kualitas classifier ROC dilihat berdasarkan akurasi dengan rentang yang diperlihatkan pada Tabel 2.1 [12]

Tabel 2.1 Nilai Kualitas Classifier

Rentang Akurasi	Klasifikasi
0.90 – 1.00	Excellent
0.80 – 0.90	Good
0.70 – 0.80	Fair
0.60 – 0.70	Poor
0.50 – 0.60	Failure

Interpretasi ROC-AUC curve dapat digunakan untuk membandingkan setiap pengklasifikasi terhadap data penelitian agar mengetahui kinerja setiap model klasifikasi prediksi terbaik. Setiap pengklasifikasi diuji dengan data testing dan data standard, kemudian dipetakan antara sensitivity dan specificity menggunakan ROC curve. Untuk pengklasifikasi optimal nilai sensitivity = 0 dalam kasus sempurna, yang menunjukkan AUC adalah 1 dan untuk situasi sub-optimal kurang dari 1. Untuk mengevaluasi kinerja kuantitatif, nilai AUC menjadi acuan pada setiap pengklasifikasi (Sultana, Haider, & Uddin, 2016).

BAB III

HASIL DAN PEMBAHASAN

Kami menggunakan dataset Iris dari Package yang telah tersedia di google colaboratory untuk melakukan prediksi jenis bunga anggrek. Disarankan anda untuk mengunduh dataset dan mengikuti langkah yang ada pada tulisan ini.

```
from sklearn.datasets import load_iris
import pandas as pd
import numpy as np

iris = load_iris()
```

ta set ini terdiri dari 3 jenis bunga anggrek: setosa, versicolor, virginica
dapat 4 buah : panjang lebar daun, panjang lebar kelopak bunga
gi data menjadi data training dan testing; contoh kali ini ambil 2 features saja

```
# Training Data
# 25 data dari setiap jenis Bunga anggrek

x1 = iris.data[ 0 : 25 ,:2]
x2 = iris.data[50 : 75 ,:2]
x3 = iris.data[100:125 ,:2]

#jadikan x1,x2,x3 sebagai data training dengan 2 features saja
x_train = np.concatenate((x1,x2,x2))

y1 = iris.target[0:25]
y2 = iris.target[50:75]
y3 = iris.target[100:125]
y_train= np.concatenate((y1,y2,y3))

# isinya apa aja
y_train

array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2])
```

- Data set ini berupa data multiclass yang terdiri dari 3 jenis bunga anggrek, yaitu sentosa, versicolor, dan virginica. Data ini terdiri dari 4 buah feature, yaitu panjang dan lebar daun, panjang dan lebar kelopak bunga.
- Data kami bagi menjadi 2, yaitu data training dan data data testing. Pada contoh ini kami ambil 2 buah feature saja dari data set. Dari iris diatas kami ambil bagian datanya, karena masing-masing bunga terdiri dari 50 bunga, dengan membagi 25 data dari setiap jenis

bunga anggrek secara acak untuk digunakan sebagai data training yang nanti modelnya akan di uji pada data testing. Data testing juga kami bagi menjadi 25 data dari setiap jenis bunga anggrek secara acak dan sisa dari data training yang belum digunakan.

Interpretasi: Berdasarkan output diatas, menampilkan isi y_train, ada 25 yang merupakan kelas 0, ada 25 yang merupakan kelas 1, dan ada 25 entry yang merupakan kelas 2.

Code untuk Data Testing

```
# Data Testing YAITU DATA SISANYA
x4 = iris.data[ 25 : 50 ,:2]
x5 = iris.data[75 : 100 ,:2]
x6 = iris.data[125:150 ,:2]
x_test = np.concatenate((x4,x5,x6))

y4 = iris.target[25:50]
y5 = iris.target[75:100]
y6 = iris.target[125:150]
y_test= np.concatenate((y4,y5,y6))

5] # Menyimpan nama kelas

class_names = iris.target_names
class_names

array(['setosa', 'versicolor', 'virginica'], dtype='<U10')

6] from sklearn.neighbors import KNeighborsClassifier
# Jarak Euclidean

#Masukkan data training ke classifier

klasifikasi = KNeighborsClassifier(n_neighbors=1, weights='uniform', p=2)
klasifikasi.fit(x_train,y_train)

KNeighborsClassifier(n_neighbors=1)

7] y_pred = klasifikasi.predict(x_test)
y_pred
# Hasil Prediksi Kelas

# ada kesalahan yang awalnya 0 ternyata 2

array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0,
       0, 0, 0, 2, 2, 1, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2,
       1, 1, 1, 1, 2, 1, 2, 1, 1, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 1, 2, 1,
       2, 1, 1, 1, 1, 1, 1, 2, 1])
```

- Sisa dari data, kami jadikan data testing, 25 data setiap bunga digabungkan menjadi data test. Kami harus mempunyai label dari data tersebut untuk menghitung accuracynya.
- Lalu kami simpan pada iris.target

- Kami menggunakan klasifikator KNN, lalu kami masukkan data training kami

Interpretasi :

- Berdasarkan output diatas, kami menamakan kelasnya yaitu setosa, versicolor dan virginica.
- Berdasarkan output diatas, terdapat hasil prediksi kelas dari data testing kami mayoritas kelas 0, namun ada kesalahan yang seharusnya kelas 0 tetapi diprediksi sebagai kelas 2, kelas 1 dan 2 cukup tercampur hasilnya yang kemungkinan memang kesalahannya cukup banyak.

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

# Gunakan CM,, dengan memberikan data y test dan y pred
# kemudian fungsi ini akan membandingkan mana yg sama dan mana yg tidak sama
cm = confusion_matrix(y_test,y_pred)
cm

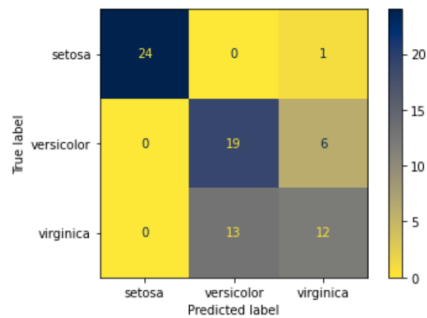
array([[24,  0,  1],
       [ 0, 19,  6],
       [ 0, 13, 12]])
```

- ▼ Ada 24 data dari kelas 0 yg diklasifikasikan secara benar dan ada 1 yg diklasifikasikan secara salah
- Ada 19 data dari kelas 1 yg diklasifikasikan secara benar dan ada 6 yg diklasifikasikan secara salah
- Ada 12 data dari kelas 2 yg diklasifikasikan secara benar dan ada 13 yg diklasifikasikan secara salah

Interpretasi :

- Terdapat 24 data dari kelas 0 yang diklasifikasikan secara benar dan ada 1 data yang diklasifikasikan secara salah.
- Terdapat 19 data dari kelas 1 yang diklasifikasikan secara benar dan ada 6 data yang diklasifikasikan secara salah.
- Terdapat 12 data dari kelas 2 yang diklasifikasikan secara benar dan ada 13 data yang diklasifikasikan secara salah.

```
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=class_names)
disp.plot(cmap='cividis_r')
plt.show()
```



Interpretasi dari Heatmap :

- Terdapat 24 data dari kelas setosa yang diklasifikasikan secara benar dan ada 1 data yang diklasifikasikan secara salah, yaitu kelas virginica.
- Terdapat 19 data dari kelas versicolor yang diklasifikasikan secara benar dan ada 6 data yang diklasifikasikan secara salah, yaitu virginica.
- Terdapat 12 data dari kelas 2 yang diklasifikasikan secara benar dan ada 13 data yang diklasifikasikan secara salah, yaitu versicolor.

Akurasi, Presisi, Recall, dan F-1 Score

```
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred, target_names= class_names))
```

	precision	recall	f1-score	support
setosa	1.00	0.96	0.98	25
versicolor	0.59	0.76	0.67	25
virginica	0.63	0.48	0.55	25
accuracy			0.73	75
macro avg	0.74	0.73	0.73	75
weighted avg	0.74	0.73	0.73	75

Akurasi 73% dari 75 data

```
[ ] [(24+19+12)/75]*100
73.33333333333333
```

Interpretasi :

Berdasarkan output diatas, data masing-masing kelas ada 25 maka tertulis di output support. Diperoleh score F1 sebesar 0.73, hal ini berarti model pada 75 data memiliki keakuratan sebesar 73%. jika dihitung manual kita akan mendapatkan accuracy dari klasifikasinya 73% sesuai dengan output classification report diatas.

Accuracy : Akurasi menjawab pertanyaan “Berapa persen jenis setiap bunga anggrek yang benar diprediksi dan Tidak benar dari keseluruhan jenis bunga anggrek?”.

Precision: Precision menjawab pertanyaan “Berapa persen jenis setiap bunga anggrek yang benar dari keseluruhan data **yang diprediksi** benar jenisnya?”

Recall : menjawab pertanyaan “Berapa persen jenis setiap bunga anggrek yang diprediksi benar jenisnya dibandingkan keseluruhan data **yang sebenarnya** jenis bunganya?”.

```
# Setosa
TP_setosa = cm[0][0]
FP_setosa = cm[0][1] + cm[0][2]
FN_setosa = cm[1][0] + cm[2][0]
TN_setosa = cm[1][1] + cm[1][2] + cm[2][1] + cm[2][2]

# Versicolor
TP_versicolor = cm[1][1]
FP_versicolor = cm[1][0] + cm[1][2]
FN_versicolor = cm[0][1] + cm[2][1]
TN_versicolor = cm[0][0] + cm[0][2] + cm[2][0] + cm[2][2]

# Virginica
TP_virginica = cm[2][2]
FP_virginica = cm[2][0] + cm[2][1]
FN_virginica = cm[0][2] + cm[1][2]
TN_virginica = cm[0][0] + cm[0][1] + cm[1][0] + cm[1][1]

# specificity Setosa
specificity_setosa = TN_setosa/float(TN_setosa+FP_setosa)

# specificity Versicolor
specificity_versicolor = TN_versicolor/float(TN_versicolor+FP_versicolor)

# specificity Virginica
specificity_virginica = TN_virginica/float(TN_virginica+FP_virginica)

print("specificity Setosa",specificity_setosa)
print("Specificity Versicolor",specificity_versicolor)
print("Specificity Virginica",specificity_virginica)

specificity Setosa 0.9803921568627451
Specificity Versicolor 0.8604651162790697
Specificity Virginica 0.7678571428571429
```

Specificity : Menjawab pertanyaan “Berapa persen jenis bunga anggrek yang benar diprediksi (True Negatif) dibandingkan dengan keseluruhan jenis bunga anggrek yang sebenarnya tidak sesuai dengan prediksi tiap jenis bunganya?”.

```

from itertools import cycle

from sklearn import svm, datasets
from sklearn.metrics import roc_curve, auc
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import label_binarize
from sklearn.multiclass import OneVsRestClassifier
from sklearn.metrics import roc_auc_score

]

# Import some data to play with
iris = datasets.load_iris()
X = iris.data
y = iris.target

# Binarize the output
y = label_binarize(y, classes=[0, 1, 2])
n_classes = y.shape[1]

# Add noisy features to make the problem harder
random_state = np.random.RandomState(0)
n_samples, n_features = X.shape
X = np.c_[X, random_state.randn(n_samples, 200 * n_features)]

```

```

# Import some data to play with
iris = datasets.load_iris()
X = iris.data
y = iris.target

# Binarize the output
y = label_binarize(y, classes=[0, 1, 2])
n_classes = y.shape[1]

# Add noisy features to make the problem harder
random_state = np.random.RandomState(0)
n_samples, n_features = X.shape
X = np.c_[X, random_state.randn(n_samples, 200 * n_features)]

# shuffle and split training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=0)

# Learn to predict each class against the other
classifier = OneVsRestClassifier(
    svm.SVC(kernel="linear", probability=True, random_state=random_state)
)
y_score = classifier.fit(X_train, y_train).decision_function(X_test)

# Compute ROC curve and ROC area for each class
fpr = dict()
tpr = dict()
roc_auc = dict()
for i in range(n_classes):
    fpr[i], tpr[i], _ = roc_curve(y_test[:, i], y_score[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])

# Compute micro-average ROC curve and ROC area
fpr["micro"], tpr["micro"], _ = roc_curve(y_test.ravel(), y_score.ravel())
roc_auc["micro"] = auc(fpr["micro"], tpr["micro"])

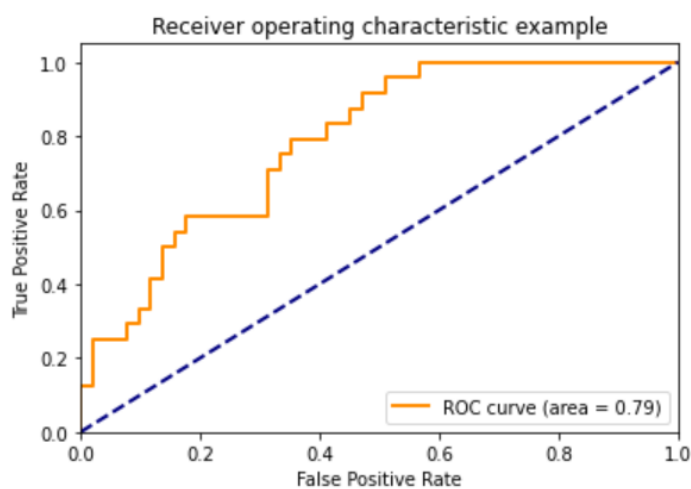
```



```

plt.figure()
lw = 2
plt.plot(
    fpr[2],
    tpr[2],
    color="darkorange",
    lw=lw,
    label="ROC curve (area = %0.2f)" % roc_auc[2],
)
plt.plot([0, 1], [0, 1], color="navy", lw=lw, linestyle="--")
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("Receiver operating characteristic example")
plt.legend(loc="lower right")
plt.show()

```



ROC merupakan kurva probabilitas yang menggambarkan TPR (True Positive Rate) terhadap FPR (False Positive Rate). The Area Under Curve (AUC) adalah ukuran kemampuan classifier untuk membedakan antara kelas dan digunakan sebagai ringkasan dari kurva ROC.

Ketika skor AUC 0.5 artinya model tersebut tidak berguna. Misalnya dalam klasifikasi apakah seseorang itu positif atau negatif, dan model memberitahu bahwa mungkin positif dan mungkin negative (50:50) maka itu tidak berguna untuk klasifikasi sesuatu.

Interpretasi :

Terlihat pada kurva ROC di atas, model yang digunakan dalam pengklasifikasian bunga anggrek memiliki skor AUC 0.79, hal ini berarti sebesar 79% model dapat mengklasifikasikan tiga jenis bunga anggrek dengan benar.

Compute macro-average ROC curve and ROC area

```
# First aggregate all false positive rates
all_fpr = np.unique(np.concatenate([fpr[i] for i in range(n_classes)]))

# Then interpolate all ROC curves at this points
mean_tpr = np.zeros_like(all_fpr)
for i in range(n_classes):
    mean_tpr += np.interp(all_fpr, fpr[i], tpr[i])

# Finally average it and compute AUC
mean_tpr /= n_classes

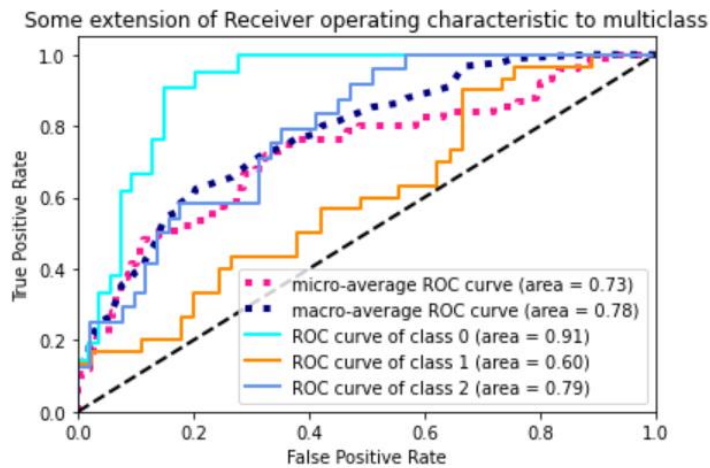
fpr["macro"] = all_fpr
tpr["macro"] = mean_tpr
roc_auc["macro"] = auc(fpr["macro"], tpr["macro"])

# Plot all ROC curves
plt.figure()
plt.plot(
    fpr["micro"],
    tpr["micro"],
    label="micro-average ROC curve (area = {0:0.2f})".format(roc_auc["micro"]),
    color="deeppink",
    linestyle=":",
    linewidth=4,
)
```

```
plt.plot(
    fpr["macro"],
    tpr["macro"],
    label="macro-average ROC curve (area = {0:0.2f})".format(roc_auc["macro"]),
    color="navy",
    linestyle=":",
    linewidth=4,
)

colors = cycle(["aqua", "darkorange", "cornflowerblue"])
for i, color in zip(range(n_classes), colors):
    plt.plot(
        fpr[i],
        tpr[i],
        color=color,
        lw=lw,
        label="ROC curve of class {0} (area = {1:0.2f})".format(i, roc_auc[i]),
    )

plt.plot([0, 1], [0, 1], "k--", lw=lw)
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("Some extension of Receiver operating characteristic to multiclass")
plt.legend(loc="lower right")
plt.show()
```



Interpretasi :

Tidak ada model yang sempurna di sini, tetapi semuanya jauh dari baseline (model yang tidak dapat digunakan (AUC 0.50)). Terlihat bahwa 79% model mampu mengklasifikasikan anggrek jenis 0, dan sebesar 91% model mampu mengklasifikasikan bunga anggrek jenis 2 namun dalam mengklasifikasikan anggrek jenis 1 model dikatakan gagal dengan kemampuan klasifikasi sebesar 60%.

Area under ROC for the multiclass problem

The `func: sklearn.metrics.roc_auc_score` function can be used for multi-class classification. The multi-class One-vs-One scheme compares every unique pairwise combination of classes. In this section, we calculate the AUC using the OvR and OvO schemes. We report a macro average, and a prevalence-weighted average.

```
[ ] y_prob = classifier.predict_proba(X_test)

macro_roc_auc_ovo = roc_auc_score(y_test, y_prob, multi_class="ovo", average="macro")
weighted_roc_auc_ovo = roc_auc_score(
    y_test, y_prob, multi_class="ovo", average="weighted")
macro_roc_auc_ovr = roc_auc_score(y_test, y_prob, multi_class="ovr", average="macro")
weighted_roc_auc_ovr = roc_auc_score(
    y_test, y_prob, multi_class="ovr", average="weighted")
print(
    "One-vs-One ROC AUC scores:\n{:.6f} (macro),\n{:.6f} "
    "(weighted by prevalence)".format(macro_roc_auc_ovo, weighted_roc_auc_ovo))
print(
    "One-vs-Rest ROC AUC scores:\n{:.6f} (macro),\n{:.6f} "
    "(weighted by prevalence)".format(macro_roc_auc_ovr, weighted_roc_auc_ovr))

One-vs-One ROC AUC scores:
0.698586 (macro),
0.665839 (weighted by prevalence)
One-vs-Rest ROC AUC scores:
0.698586 (macro),
0.665839 (weighted by prevalence)
```

AUC mengukur kinerja diskriminatif dengan memperkirakan probabilitas output dari sampel yang dipilih secara acak dari populasi positif atau negative.

Interpretasi :

- Dengan macro average diperoleh skor sebesar 0.698586
- Dengan prevalence-weighted average diperoleh skor sebesar 0.665839

Dengan metode macro average ataupun prevalence-weighted average diperoleh skor AUC model yang lemah (0.60-0.70). maka dapat dikatakan bahwa model ini kurang baik dalam mengklasifikasi ketiga jenis anggrek (sentosa, versicolor, dan virginica).

BAB IV

PENUTUP

4.1. Kesimpulan

1. Implementasi Supervised Learning Classification dengan melakukan klasifikasi jenis bunga terhadap data bunga iris menghasilkan 3 jenis kelas, yaitu setosa, versicolor dan virginica disertai dengan 2 featuranya. Diperoleh model yang cukup baik, berikut clasification report atau matrix evaluasi (Precision, Recall, F- Score dan Specificity) untuk setiap jenis bunga anggrek
 - Setosa : Dengan Precision 100%, Recall 96% , F-1 Score 98% dan Specificity 98%, semakin besar precision maka recall kecil dari nilai precision
 - Versicolor :Dengan Precision 59%, Recall 76% , F-1 Score 67% dan Specificity 86%, semakin besar recall maka precision kecil dari nilai recall
 - Virginica : Dengan Precision 63%, Recall 48% , F-1 Score 55% dan Specificity 76%, semakin besar precision maka recall kecil dari nilai precision.
2. Akurasi yang diperoleh dari model dalam mengklasifikasikan jenis bunga anggrek adalah 73% yang artinya rasio prediksi benar untuk setiap jenis bunganya dari keseluruhan data sebesar 73%

DAFTAR PUSTAKA

1. M. Ramageri, 2010, *Data Mining Techniques and Applications*. Indian J. Comput. Sci.Eng., Vol. 1, No. 4, pp. 301–305.
2. M. Karim and R. M.Rahman, 2013, *Decision Tree and Naïve Bayes Algorithm for classification and Generation of Actionable Knowledge for Direct Marketing*, J. Softw. Eng. Appl., Vol. 6, pp. 196–206.
3. Husein, Saddam. 2022. *Supervised dan Unsupervised Learning: Penjelasan, Perbedaan dan Contoh*. <https://geospasialis.com/supervised-learning-dan-unsupervised-learning/> Diakses pada 16 April 2022.
4. Sartika, Dewi, Sensuse, Dana Indra. 2017. *Perbandingan Algoritma Klasifikasi Naive Bayes, Nearest Neighbour, dan Decision Tree pada Studi Kasus Pengambilan Keputusan Pemilihan Pola Pakaian*. Lembaga Penelitian dan Pengabdian pada Masyarakat (LPPM) STMIK Global Informatika MDP.
5. Sarang Narkhede. *Understanding Confusion Matrix*. May 2018. <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62> Diakses pada 16 April 2022.
6. Jianfeng Xu, Yuanjian Zhang, Duogian Miao. *Three-way confusion matrix for classification: A measure driven view*. Information Sciences Volume 507, January 2020. Elsevier. <https://doi.org/10.1016/j.ins.2019.06.064>.
7. Lingga Aji Andika, Pratiwi Amalia Nur Azizah, dan Respatiwan. 2019. *Analisis Sentimen Masyarakat terhadap Hasil Quick Count Pemilihan Presiden Indonesia 2019 pada Media Sosial Twitter Menggunakan Metode Naive Bayes Classifier*. Indonesian Journal of Applied Statistics Volume 2 No. 1 May 2019 .
8. Željko Đ. Vujović. 2021. Classification Model Evaluation Metrics. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 12, No. 6.
9. Kwetishe Joro Danjuma. 2015. *Performance Evaluation of Machine Learning Algorithms in Post-operative Life Expectancy in the Lung Cancer Patients*. IJCSI International Journal of Computer Science Issues, Volume 12, Issue 2, March 2015.
10. Aziz Taha, Abdel. 2015. *Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool*. BMC Medical Imaging. 15 (29): 1–28.
11. Dr. Niraj Kumar. 2020. *How to Compute Average F1, Macro F1 and Micro F1 for Multi-Class Classification* <https://www.youtube.com/watch?v=L2tBh63ggt0>. Diakses pada 16 April 2022.

12. Gorunescu, F. (2011). *Data Mining : Concepts, models and techniques*. Verlag Berlin Heidelberg: Springer.
13. Vercellis, C. (2006). *Business Intelligence: Data mining and Optimization for Decision Making*. Milano, Italy: Wiley.
14. Sultana, M., Haider, A., & Uddin, M. S. (2016). *Analysis of Data Mining Techniques for Heart Disease Prediction*. 2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT).