***PROJECT* UJIAN AKHIR SEMESTER**

PENGANTAR DATA SAINS

PERBANDINGAN ALGORITMA *MULTI-LAYER PERCEPTRON* (MLP) DALAM KLASIFIKASI STUNTING PADA ANAK DENGAN PENDEKATAN GRIDSEARCHCV DAN HYPEROPT

Aulia Mirfah Setyo Ayu Damayanti

22/493337/PA/21178

Syifa Azzahra Susilo

22/496431/PA/21342

Rhizka Febrianty Aisyah Dewi

22/503643/PA/21625

DOSEN PENGAMPU

Drs. Danardono, MPH., Ph.D.

Mohammad Fahruli Wahyujati, S.Si., M.Si.

PROGRAM STUDI STATISTIKA

DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS GADJAH MADA

YOGYAKARTA

2024

ABSTRAK

Stunting adalah kondisi kronis yang terjadi pada anak-anak akibat kurangnya asupan nutrisi dalam jangka panjang, mengakibatkan gangguan pertumbuhan fisik dan perkembangan kognitif. Stunting juga berkaitan dengan risiko kesehatan serius di masa depan, seperti diabetes dan penyakit jantung. Di Indonesia, masalah stunting sangat signifikan, memengaruhi jutaan anak dan menunjukkan kesenjangan dalam akses gizi, layanan kesehatan, sanitasi, dan pendidikan, terutama di daerah pedesaan dan keluarga berpenghasilan rendah. Penelitian ini bertujuan mengembangkan model yang akurat dan efisien untuk mengidentifikasi anak-anak yang berisiko stunting menggunakan algoritma Multi-Layer Perceptron (MLP). Data penelitian diperoleh dari Kaggle, terdiri dari 10000 baris data dengan 8 variabel. Model MLP dilatih menggunakan dua pendekatan *tuning hyperparameter*, yaitu GridSearchCV dan Hyperopt. Hasil menunjukkan bahwa Hyperopt memberikan akurasi tertinggi sebesar 80.1%, dibandingkan dengan GridSearchCV yang mencapai 77.7%. Analisis *feature importance* menunjukkan bahwa panjang lahir, berat lahir, dan usia merupakan faktor-faktor utama dalam prediksi stunting. Penelitian ini memberikan kontribusi penting dalam upaya mengatasi stunting di Indonesia, menawarkan model yang dapat membantu pemangku kepentingan dalam merancang intervensi yang lebih efektif dan tepat sasaran. Dengan penggunaan teknologi dan metode optimasi *hyperparameter* yang canggih, penelitian ini mendukung peningkatan kualitas hidup anak-anak Indonesia menuju generasi yang lebih sehat dan produktif.

Kata Kunci : *Stunting*, MLP, GridSearchCV, Hyperopt, *Hyperparameter*

1. PENDAHULUAN

1.1 Latar Belakang

Stunting merupakan kondisi kronis yang terjadi pada anak-anak ketika mereka tidak mendapatkan nutrisi yang cukup dalam jangka waktu yang Panjang. Hal ini dapat mengakibatkan pertumbuhan tinggi badan mereka terganggu dan lebih pendek dibandingkan dengan usia mereka. Stunting bukan hanya masalah pertumbuhan fisik yang terhambat, tetapi juga merupakan indikator dari berbagai masalah kesehatan lainnya yang lebih serius. Anak-anak yang mengalami stunting cenderung memiliki perkembangan kognitif yang buruk, sistem kekebalan tubuh yang lemah, dan risiko lebih tinggi terhadap penyakit kronis seperti diabetes dan penyakit jantung di kemudian hari.

Di Indonesia, stunting telah menjadi permasalahan serius yang mempengaruhi jutaan anak. Menurut data dari UNICEF dan WHO, Indonesia berada pada peringkat ke-27 dari 154 negara dengan tingkat prevalensi stunting yang tinggi, dan di kawasan Asia, Indonesia menempati peringkat ke-5. Prevalensi stunting yang tinggi di Indonesia mencerminkan adanya masalah mendasar dalam hal akses terhadap gizi yang cukup, pelayanan kesehatan, sanitasi, dan pendidikan. Anak-anak yang tinggal di daerah pedesaan dan keluarga dengan tingkat ekonomi rendah lebih rentan terhadap stunting. Seiring dengan peningkatan kesadaran masyarakat dan pemerintah terhadap pentingnya mengatasi stunting, berbagai upaya telah dilakukan untuk mengurangi prevalensi stunting di Indonesia. Namun, masih banyak tantangan yang harus dihadapi, termasuk kebutuhan akan alat dan metode yang efektif untuk mengidentifikasi anak-anak yang berisiko stunting serta untuk mengevaluasi dampak dari intervensi yang dilakukan.

Melalui penelitian ini, diharapkan dapat diperoleh model yang lebih akurat dan efisien dalam mengidentifikasi anak-anak yang berisiko stunting. Hasil dari penelitian ini dapat digunakan oleh pemangku kepentingan, termasuk pemerintah, dan penyedia layanan kesehatan, untuk merancang intervensi yang lebih tepat sasaran dan efektif dalam mengurangi prevalensi stunting di Indonesia. Dengan demikian, kontribusi penelitian ini diharapkan dapat membantu meningkatkan kualitas hidup anak-anak Indonesia dan mewujudkan generasi yang lebih sehat dan produktif di masa depan.

1.2 Tujuan Penelitian

1. Melakukan dan menilai performa algoritma *Multi-Layer Percepton* (MLP) dalam klasifikasi stunting pada anak.
2. Melakukan perbandingan antara pendekatan GridSearchCV dan Hyperopt dalam optimasi hyperparameter untuk model Mukti-Layer Percepton (MLP).
3. Mengidentifikasi model *Multi-Layer Percepton* (MLP) dengan kombinasi hyperparameter yang optimal yang memberikan akurasi tertinggi dalam klasifikasi stunting.
4. Membantu pemerintah dan organisasi lainnya dalam membuat program yang lebih efektif untuk mengurangi stunting dengan menggunakan model yang akurat dan terbaik.

2. DATA DAN METODE PENELITIAN

2.1 Data

Data yang digunakan dalam penelitian ini adalah data sekunder yang diperoleh dari website kaggle.com. Dataset yang digunakan merupakan data kesehatan yang berisi informasi relevan untuk menganalisis faktor-faktor yang memengaruhi stunting pada balita. Dataset ini terdiri dari 10000 data dengan 8 variabel yang dijelaskan sebagai berikut:

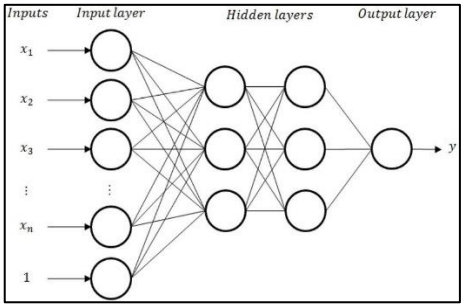
- **Gender (kategorik):** jenis kelamin (Male, Female)
- **Age (numerik):** umur badan bayi saat ini
- **Birth Weight (numerik):** berat badan balita saat lahir
- **Birth Length (numerik):** panjang badan balita saat lahir
- **Body Weight (numerik):** berat badan balita saat ini
- **Body Length (numerik):** panjang badan balita saat ini
- **Breastfeeding (kategorik):** status pemberian ASI pada bayi (No, Yes)
- **Stunting (kategorik):** status stunting pada balita (No, Yes)

2.2 Multi-Layer Perceptron (MLP)

Metode klasifikasi Multi-Layer Perceptron (MLP) merupakan salah satu jenis dari algoritma jaringan saraf tiruan yang mengadopsi cara kerja jaringan saraf pada mahluk hidup. Algoritma ini terkenal handal karena proses pembelajaran yang

mampu dilakukan secara terarah. Pembelajaran algoritma ini dilakukan dengan memperbarui bobot balik (backpropagation). Penetapan bobot yang optimal akan menghasilkan hasil klasifikasi yang tepat. MLP terdiri dari sistem yang sederhana saling menghubungkan jaringan atau node.

Node tersebut dihubungkan oleh bobot dan unit output yang merupakan fungsi penjumlahan dari input ke node dimodifikasi oleh transfer non-linear sederhana, atau aktifasi. MLP merupakan algoritma supervised yang membutuhkan proses pembelajaran untuk menentukan bobot yang optimal yang digunakan dalam proses pengujian data. Selama proses pelatihan, MLP secara berkala terjadi perubahan bobot dengan data latih pada jaringan sampai map input-output yang diinginkan terjadi.



Gambar 1. Contoh Arsitektur Multi-Layer Perceptron

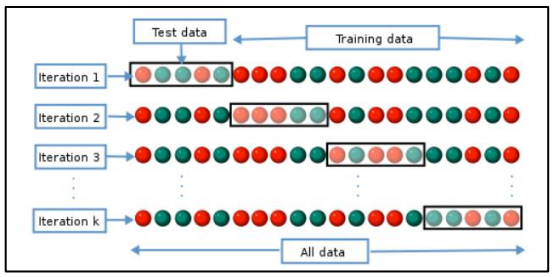
Algoritma MLP dimulai dengan inisialisasi bobot secara acak kecil, kemudian sinyal diteruskan dari unit input ke unit tersembunyi hingga unit keluaran (forward propagation). Gradien error dihitung (backpropagation) dan bobot diperbarui dengan algoritma seperti gradient descent. Proses ini diulangi hingga mencapai iterasi maksimum atau error minimum.

2.3 *Grid Search Cross-Validation* (GridSearchCV)

GridSearchCV adalah fungsi dalam pustaka scikit-learn yang digunakan untuk mengidentifikasi parameter optimal dari sebuah pengklasifikasi sehingga model dapat secara akurat memprediksi data yang tidak berlabel [Siji George C. G. & B. Sumathi, 2020]. GridSearchCV menyederhanakan proses penyetelan model dengan melakukan pencarian menyeluruh pada ruang parameter yang ditentukan, mencoba semua kombinasi yang mungkin dari hyperparameter. Metode ini membantu

memastikan model beroperasi secara optimal untuk tugas spesifik tanpa menimbulkan biaya komputasi yang berlebihan. Selain itu, GridSearchCV juga dapat membantu mencegah overfitting karena alat ini akan memilih hyperparameter yang menghasilkan kinerja generalisasi terbaik pada set pengujian.

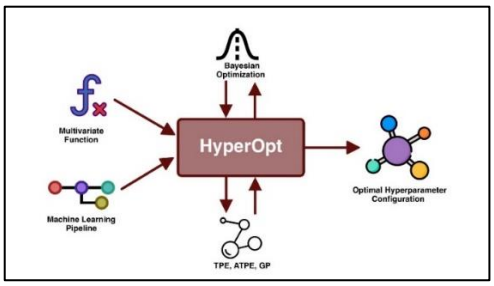
GridSearchCV bekerja dengan mendefinisikan kisi-kisi hyperparameter, kemudian secara sistematis melatih dan mengevaluasi model pembelajaran mesin untuk setiap kombinasi hyperparameter. Proses pelatihan dan evaluasi model untuk setiap kombinasi disebut validasi silang (cross-validation). Dalam validasi silang, data dibagi menjadi beberapa subset, model dilatih pada sebagian subset, dan diuji pada subset yang tersisa. Proses ini diulang untuk memastikan hasil yang lebih andal. Kumpulan hyperparameter terbaik kemudian dipilih berdasarkan metrik evaluasi.



Gambar 2. *Cross-Validation*

Untuk menggunakan GridSearchCV, pertama-tama identifikasi hyperparameter yang ingin dioptimalkan serta rentang nilainya. Kemudian, definisikan model machine learning yang akan digunakan beserta parameternya. Terakhir, tentukan metrik evaluasi seperti akurasi, presisi, recall, atau F1-score untuk mengevaluasi performa berbagai kombinasi hyperparameter.

2.4 Hyperopt



Gambar 3. Arsitektur Perwakilan Hyperopt

HyperOpt adalah pustaka python *open-source* yang dibuat oleh James Bergstra pada tahun 2011. HyperOpt adalah fitur yang memungkinkan otomatisasi pencarian hyperparameter optimal dari model pembelajaran mesin. HyperOpt didasarkan pada Bayesian Optimization yang didukung oleh metodologi SMBO yang diadaptasi untuk bekerja dengan algoritma yang berbeda seperti: *Tree of Parzen Estimators* (PE), *Adaptive Tree of Parzen Estimators* (ATPE) dan *Gaussian Processes* (GP).

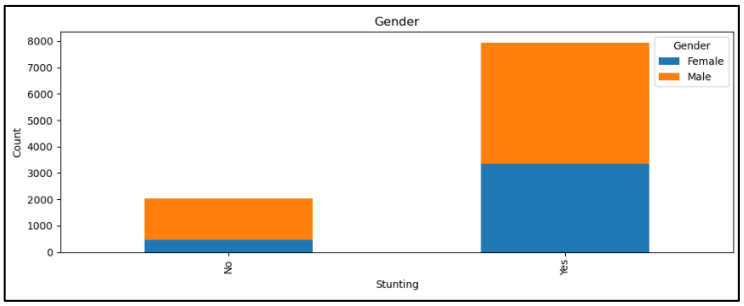
Pendekatan Pengoptimalan Bayesian berfokus pada model probabilitas P (skor | konfigurasi), yang diperbarui melalui proses berulang untuk menanyakan riwayat " H " dari (skor, konfigurasi) yang tujuannya adalah memaksimalkan skor dengan konfigurasi "c". HyperOpt mengambil Bayesian Optimization sebagai premisnya dengan membuat beberapa variasi dalam proses pengambilan sampel, definisi dan mempersempit ruang pencarian dan algoritma untuk memaksimalkan model probabilitas HyperOpt membutuhkan 4 komponen penting untuk pengoptimalan hyperparameter: ruang pencarian, fungsi kerugian, algoritma pengoptimalan, dan *database* untuk menyimpan riwayat (skor, konfigurasi). Ruang pencarian akan ditentukan oleh fungsi kontinu dan cembung. Fungsi kerugian adalah fungsi yang perlu dioptimalkan, yang diperoleh dengan mengevaluasi model dengan konfigurasi “c”. The optimasi algoritma didasarkan pada SMBO metodologi dengan varian yang diberikan oleh GP, TPE, dan ATPE algoritma. The “H” Database toko satu set tupel (s inti, konfigurasi) yang diperoleh melalui iterasi dari algoritma optimasi.

3. HASIL DAN DISKUSI

3.1 Exploratory Data Analysis (EDA)

Tahap EDA dilakukan untuk memperoleh informasi yang tersimpan dalam data. Berikut merupakan EDA yang dilakukan pada dataset yang diberikan.

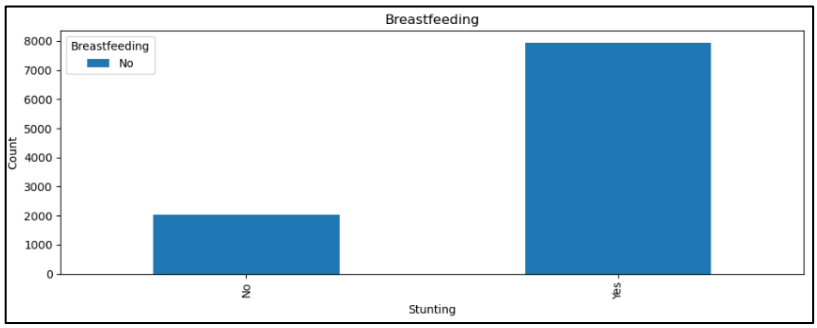
Barchart Gender Berdasarkan Stunting



Gambar 4. Barchart Gender Berdasarkan Stunting

Berdasarkan grafik batang di atas, terlihat bahwa terdapat lebih banyak anak yang mengalami stunting (Yes) dibandingkan yang tidak mengalami stunting (No). Baik pada kategori "No" maupun "Yes", jumlah anak laki-laki (warna oranye) lebih tinggi dibandingkan jumlah anak perempuan (warna biru). Perbedaan jumlah antara laki-laki dan perempuan lebih mencolok pada kategori "No" (stunting) dibandingkan pada kategori "Yes".

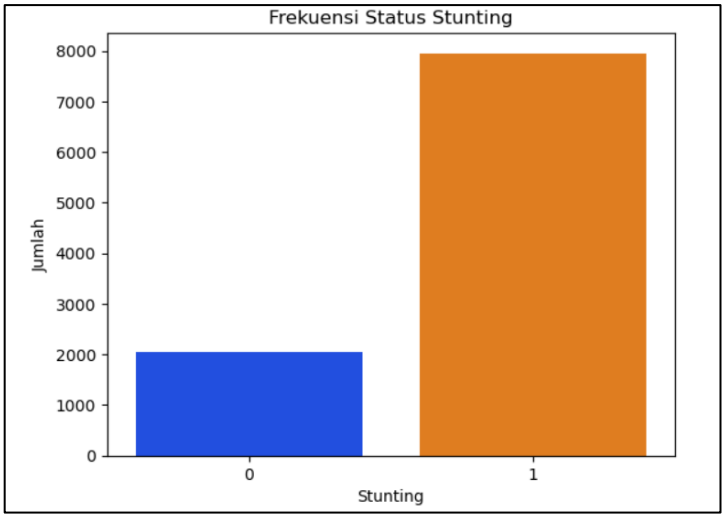
Barchart Breastfeeding Berdasarkan Stunting



Gambar 5. Barchart Breastfeeding Berdasarkan Stunting

Berdasarkan grafik batang di atas, terlihat bahwa anak yang mengalami stunting (Yes) lebih banyak dibandingkan yang tidak mengalami stunting (No). Semua data yang ditampilkan dalam grafik ini menunjukkan bahwa semua anak dalam sampel, baik yang mengalami stunting (Yes) maupun tidak mengalami stunting (NO), tidak mendapatkan ASI (Breastfeeding = No). Data ini menunjukkan adanya korelasi antara anak yang tidak mendapatkan ASI dengan angka stunting.

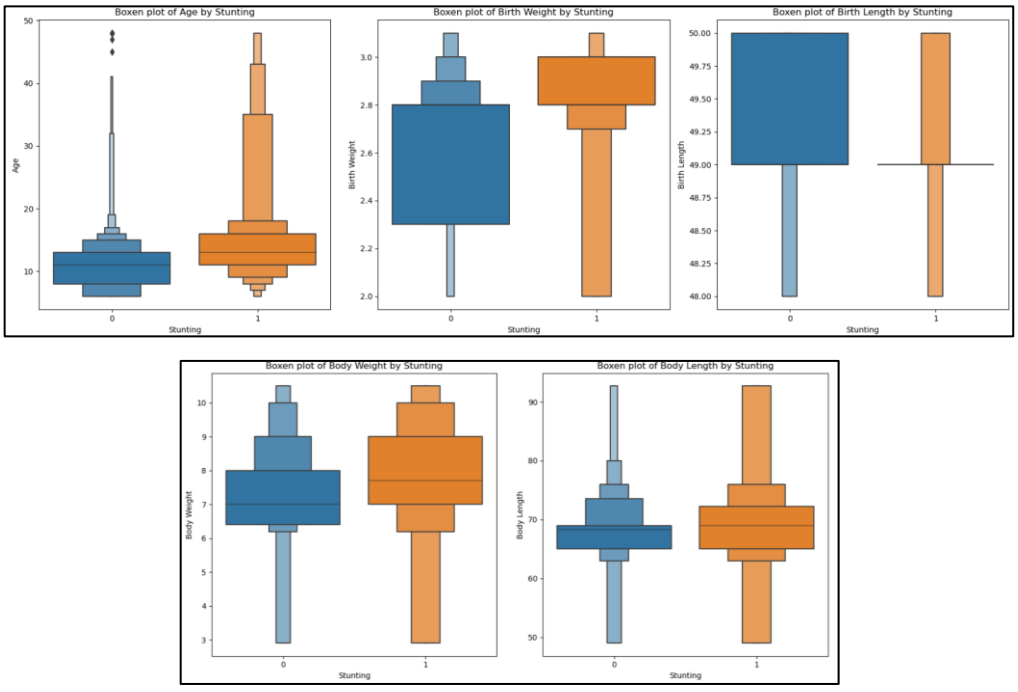
Barchart Frekuensi Status Stunting



Gambar 6. Barchart Breastfeeding Berdasarkan Stunting

Berdasarkan grafik batang di atas, terlihat adanya perbedaan yang signifikan antara data anak yang tidak mengalami stunting (0) dan yang mengalami stunting (1). Terdapat sekitar 2000 anak yang tidak mengalami stunting dan sekitar 8000 anak yang mengalami stunting dari total 10000 anak.

Boxen plot Variabel Numerik Berdasarkan Stunting



Gambar 7. Boxen plot Variabel Numerik Berdasarkan Stunting

a. ***Boxen plot of Age by Stunting*** **(Usia berdasarkan Stunting)**

- Stunting 0 (Tidak Stunting): Plot menunjukkan bahwa distribusi usia kelompok tanpa stunting memiliki rentang yang lebih luas dengan beberapa outliers di atas 40 bulan. Median usia berada di sekitar 10 bulan, dan kuartil atas sekitar 15 bulan.
- Stunting 1 (Stunting): Plot menunjukkan distribusi usia yang lebih tersebar dan terpusat di usia yang lebih tinggi dibandingkan kelompok tanpa stunting. Median usia sekitar 20 bulan, dengan rentang antar kuartil lebih besar.
- Sehingga dapat disimpulkan bahwa semakin bertambahnya umur anak, maka semakin tinggi peluang untuk mengalami stunting.

b. ***Boxen plot of Birth Weight by Stunting*** **(Berat Lahir berdasarkan Stunting)**

- Stunting 0 (Tidak Stunting): Distribusi berat lahir pada kelompok tanpa stunting menunjukkan bahwa berat lahir median sekitar 2.8 kg dengan rentang kuartil yang lebih lebar di bagian atas.
- Stunting 1 (Stunting): Distribusi berat lahir pada kelompok dengan stunting menunjukkan median yang sedikit lebih rendah dari kelompok tanpa stunting, dengan rentang kuartil lebih besar di bagian bawah.
- Sehingga dapat disimpulkan bahwa semakin rendah berat badan lahir, maka semakin tinggi kemungkinan untuk mengalami stunting.

c. ***Boxen plot of Birth Length by Stunting*** **(Panjang Lahir berdasarkan Stunting)**

- Stunting 0 (Tidak Stunting): Distribusi panjang lahir pada kelompok tanpa stunting menunjukkan median sekitar 49 cm, dengan rentang yang lebih besar ke arah bawah.
- Stunting 1 (Stunting): Distribusi panjang lahir pada kelompok dengan stunting menunjukkan median yang hampir sama dengan kelompok tanpa stunting, tetapi dengan variasi yang lebih besar di bagian bawah.
- Sehingga dapat disimpulkan bahwa panjang lahir anak cukup memengaruhi status stunting anak, dimana semakin rendah panjang lahir, maka semakin tinggi kemungkinan untuk mengalami stunting.

d. *Boxen plot of Body Weight by Stunting* (Berat Badan berdasarkan Stunting)

- Stunting 0 (Tidak Stunting): Distribusi berat badan pada kelompok tanpa stunting menunjukkan median sekitar 7 kg, dengan variasi yang lebih kecil ke arah bawah.
- Stunting 1 (Stunting): Distribusi berat badan pada kelompok dengan stunting menunjukkan median yang sedikit lebih tinggi dibandingkan kelompok tanpa stunting, dengan variasi yang lebih besar di bagian bawah.
- Sehingga dapat disimpulkan bahwa beberapa anak yang memiliki berat badan di bawah rata-rata cenderung untuk mengalami stunting.

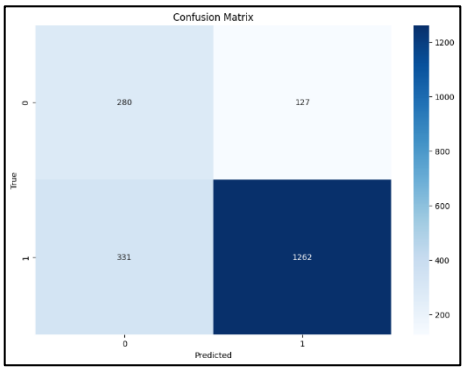
e. *Boxen plot of Body Length by Stunting* (Panjang Badan berdasarkan Stunting)

- Stunting 0 (Tidak Stunting): Distribusi panjang badan pada kelompok tanpa stunting menunjukkan median sekitar 70 cm, dengan rentang kuartil yang lebih besar di bagian atas.
- Stunting 1 (Stunting): Distribusi panjang badan pada kelompok dengan stunting menunjukkan median yang sedikit lebih rendah dari kelompok tanpa stunting, dengan rentang kuartil yang lebih besar di bagian bawah.
- Sehingga dapat disimpulkan bahwa anak-anak dengan stunting cenderung memiliki panjang badan yang sedikit lebih rendah dibandingkan anak-anak tanpa stunting.

3.2 Hasil MLP

Pada penelitian ini, algoritma model Multi-Layer Perceptron (MLP) dilatih dengan menggunakan konfigurasi hyperparameter yang optimal, yaitu hidden_layer_sizes dengan 100 neuron dan max_iter sebesar 300. Model ini menunjukkan akurasi sebesar 77.1%, yang berarti model mampu mengklasifikasikan data dengan benar sebesar 77.1% dari seluruh data uji. Hasil ini menunjukkan performa yang cukup baik dalam mengklasifikasikan data stunting anak. Secara keseluruhan, model ini mencapai accuracy sebesar 0.77, macro average untuk precision sebesar 0.68, recall sebesar 0.74, dan f1-score sebesar 0.70, sedangkan weighted average untuk precision sebesar 0.82, recall sebesar 0.77, dan f1-score sebesar 0.79, berdasarkan total 2000 sampel data uji.

Hasil evaluasi menunjukkan bahwa model MLP memiliki performa yang cukup baik sebagai langkah awal, namun mungkin belum optimal. Hal ini dapat disebabkan oleh distribusi data yang tidak seimbang. Untuk meningkatkan performa model, beberapa langkah dilakukan seperti tuning hyperparameter menggunakan GridSearchCV dan Hyperopt.



Gambar 8. Confussion Matrix MLP

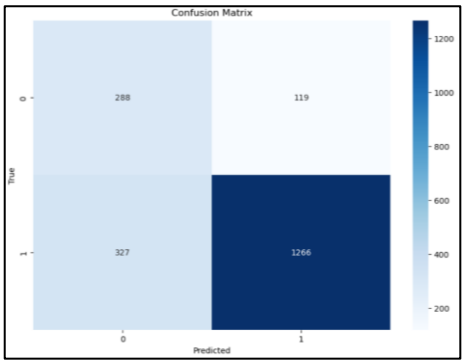
3.3 Hasil GridSearchCV

Penelitian ini memberikan hasil berupa tingkat akurasi yang diperoleh dari pengujian yang telah dilakukan dengan tujuan untuk menguji akurasi dan kinerja metode MLP dengan menerapkan tuning hyperparameter menggunakan GridSearchCV. Nilai konfigurasi *tuning hyperparameter* untuk mendapatkan nilai akurasi optimal dapat dilihat pada Tabel 3.

Tabel 1. Konfigurasi Tuning Hyperparameter GridSearchCV

Hyperparameter	Configuration	Best Parameters
hidden_layer_sizes	kombinasi ukuran lapisan tersembunyi (10,), (20,), dan (10, 10)	(20,) Satu lapisan tersembunyi dengan 20 neuron
activation	tanh atau relu	relu
solver	sgd atau adam	adam
alpha	antara 0,0001 dan 0,05	0,05
learning_rate	constant atau adaptive	constant
max_iter	100	100

Model terbaik yang ditemukan menggunakan *hyperparameter* GridsearchCV memiliki parameter sebagai berikut: fungsi aktivasi 'relu', parameter alpha sebesar 0,05, satu lapisan tersembunyi dengan 20 neuron (hidden_layer_sizes: (20,)), strategi pembelajaran 'constant', dan algoritma optimisasi 'adam'. Model ini mencapai akurasi sebesar 77,7% pada data uji, menunjukkan performa yang cukup baik dalam memprediksi hasil dengan parameter yang dioptimalkan. Secara keseluruhan, model ini mencapai *accuracy* sebesar 0,78, *macro average* untuk *precision* sebesar 0,69, *recall* sebesar 0,75, dan *f1-score* sebesar 0,71, sedangkan *weighted average* untuk *precision* sebesar 0,82, *recall* sebesar 0,78, dan *f1-score* sebesar 0,79, berdasarkan total 2000 sampel data uji.



Gambar 9. Confussion Matrix dengan GridSearchCV

3.4 Hasil Hyperopt

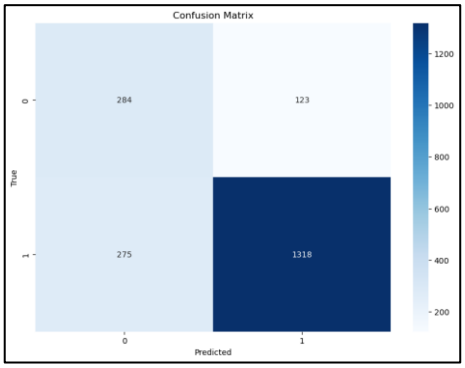
Penelitian ini memberikan hasil berupa tingkat akurasi yang diperoleh dari pengujian yang telah dilakukan dengan tujuan untuk menguji akurasi dan kinerja metode MLP dengan menerapkan tuning hyperparameter menggunakan Hyperopt. Nilai konfigurasi tuning hyperparameter untuk mendapatkan nilai akurasi optimal dapat dilihat pada Tabel 3.

Tabel 2. Konfigurasi Tuning Hyperparameter

Hyperparameter	Configuration	Best Parameters
hidden_layer_sizes	antara 10 dan 100 neuron	95 neuron
activation	tanh atau relu	tanh
alpha	diambil dari distribusi log-uniform antara $1e^{-5}$ dan $1e^{-2}$.	0.002

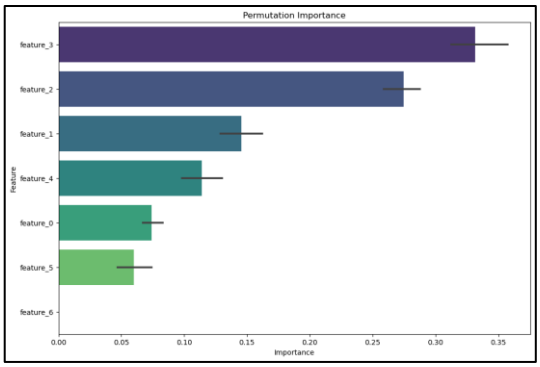
max_iter	100	100

Model terbaik yang ditemukan menggunakan Hyperopt memiliki parameter sebagai berikut: fungsi aktivasi 'tanh' (activation: 0), parameter regulasi L2 alpha sebesar 0.0020107901745201853, satu lapisan tersembunyi dengan 95 neuron (hidden_layer_sizes: 95.0 setelah dikonversi ke integer), strategi pembelajaran 'adaptive' (learning_rate: 1), dan algoritma optimisasi 'adam' (solver: 1). Model ini mencapai akurasi sebesar 80.1% pada data uji, menunjukkan performa yang cukup baik dalam memprediksi hasil dengan parameter yang dioptimalkan. Secara keseluruhan, model ini mencapai accuracy sebesar 0.80, macro average untuk precision sebesar 0.71, recall sebesar 0.76, dan f1-score sebesar 0.73, sedangkan weighted average untuk precision sebesar 0.83, recall sebesar 0.80, dan f1-score sebesar 0.81, berdasarkan total 2000 sampel data uji.



Gambar 10. Confussion Matrix dengan Hyperopt

3.5 Feature Importance



Gambar 11. Feature Importance

Tiga variabel teratas yang memiliki feature importance tertinggi adalah Birth Length (Fitur_3), Birth Weight (Fitur_2), dan Age (Fitur_1). Analisis feature importance menunjukkan bahwa Birth Length (Fitur_3) memiliki kontribusi tertinggi terhadap prediksi model dengan nilai importance sebesar 0.331772, menandakan bahwa panjang saat lahir anak sangat signifikan dalam menentukan kemungkinan stunting. Birth Weight (Fitur_2) menempati peringkat kedua dengan nilai 0.274671, menunjukkan pengaruh yang kuat meskipun sedikit lebih rendah dari Birth Length. Sementara itu, Age (Fitur_1) memiliki kontribusi paling rendah dengan nilai 0.145534, menunjukkan bahwa usia mungkin memiliki pengaruh yang lebih kecil dalam prediksi stunting dibandingkan Birth Length dan Birth Weight. Informasi ini penting untuk memahami faktor-faktor yang paling berpengaruh dalam status stunting anak, yang dapat digunakan untuk mendukung keputusan intervensi kesehatan yang lebih efektif.

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Tabel 3. Evaluasi Akurasi

MLP	MLP GridSearchCV	MLP Hyperopt
0,771	0,777	0,801

Studi ini membandingkan efektivitas algoritma *Multi-Layer Perceptron* (MLP) dalam klasifikasi stunting pada anak menggunakan dua pendekatan tuning hyperparameter, yaitu GridSearchCV dan Hyperopt. Data yang digunakan berasal dari dataset stunting Indonesia dari platform Kaggle, terdiri dari 8 variabel dan 10,000 data dengan variabel dependen `Stunting` dan variabel independen seperti `Gender`, `Age`, `Birth Weight`, `Birth Length`, `Body Weight`, `Body Length`, dan `Breastfeeding`. Hasil penelitian menunjukkan bahwa pendekatan tuning hyperparameter dengan Hyperopt menghasilkan model dengan akurasi tertinggi, mencapai 0,801, dibandingkan dengan GridSearchCV yang mencapai 0,789. Ini menunjukkan bahwa penggunaan metode optimasi hyperparameter yang lebih canggih seperti Hyperopt dapat meningkatkan kinerja dan akurasi prediksi stunting pada anak secara signifikan.

4.2 Saran

Berdasarkan temuan ini, peneliti merekomendasikan beberapa saran untuk penelitian dan kebijakan selanjutnya. Pertama, peneliti menyarankan untuk melakukan eksplorasi lebih lanjut terhadap berbagai kombinasi hyperparameter menggunakan Hyperopt atau teknik optimasi hyperparameter lainnya untuk memperbaiki performa model lebih lanjut. Kedua, disarankan untuk memperluas cakupan data dengan menambahkan variabel-variabel tambahan seperti faktor lingkungan atau data gizi yang lebih detail untuk meningkatkan akurasi prediksi stunting pada anak. Ketiga, peneliti merekomendasikan implementasi hasil penelitian ini dalam upaya penanggulangan stunting di Indonesia. Rekomendasi ini meliputi pemanfaatan teknologi deep learning sebagai alat bantu diagnosis dini, yang dapat meningkatkan efektivitas intervensi dan kebijakan kesehatan masyarakat yang relevan. Penelitian ini memberikan sumbangan penting dalam pemahaman dan penanganan stunting sebagai masalah kesehatan masyarakat di Indonesia, menyoroti pentingnya penggunaan teknologi dan metodologi yang tepat untuk mencapai tujuan tersebut.

DAFTAR PUSTAKA

Agustina, N. (2022, September 13). Apa itu Stunting. *Yankes Kemkes*. Retrieved from https://yankes.kemkes.go.id/view_artikel/1516/apa-itu-stunting

Lonang, S. (2022). Klasifikasi Status Stunting Pada Balita Menggunakan K-Nearest Neighbor Dengan Feature Selection Backward Elimination. *Jurnal Media Informatika Budidarma*, 6(1), 49-56. https://doi.org/10.30865/mib.v5i1.2293

López, F. (2021, January 24). *HyperOpt: Hyperparameter Tuning based on Bayesian Optimization*. Towards Data Science. Retrieved from https://towardsdatascience.com/hyperopt-hyperparameter-tuning-based-on-bayesian-optimization-7fa32dffaf29

LAMPIRAN

Berikut dilampirkan mengenai sumber data yang digunakan pada penelitian ini dan sintaks analisis dengan menggunakan bahasa pemrograman Python.

https://www.kaggle.com/datasets/harnelia/faktor-stunting

sintaks Python (terlampir)

project-pds-1

July 2, 2024

[42]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.base import BaseEstimator, ClassifierMixin
from sklearn.cluster import KMeans
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler
```

1 Prepare Dataset

[43]:
```
df = pd.read_csv("Stunting_Dataset.csv")
df
```

[43]:

	Gender	Age	Birth Weight	Birth Length	Body Weight	Body Length	Breastfeeding	Stunting
0	Male	17	3.0	49	10.0	72.2	No	No
1	Female	11	2.9	49	2.9	65.0	No	Yes
2	Male	16	2.9	49	8.5	72.2	No	Yes
3	Male	31	2.8	49	6.4	63.0	No	Yes
4	Male	15	3.1	49	10.5	49.0	No	Yes
…	…	…	…	…	…	…	…	…
9995	Male	15	3.0	49	9.0	63.0	No	Yes
9996	Female	12	2.8	48	7.7	63.0	No	No
9997	Male	16	2.8	49	7.7	49.0	No	Yes
9998	Male	14	2.8	49	10.0	69.0	No	Yes
9999	Female	10	3.0	49	7.7	80.0	No	Yes

```
9997            No       No
9998            No      Yes
9999            No      Yes

[10000 rows x 8 columns]
```

2 EDA

[44]:
```
df.info()
```

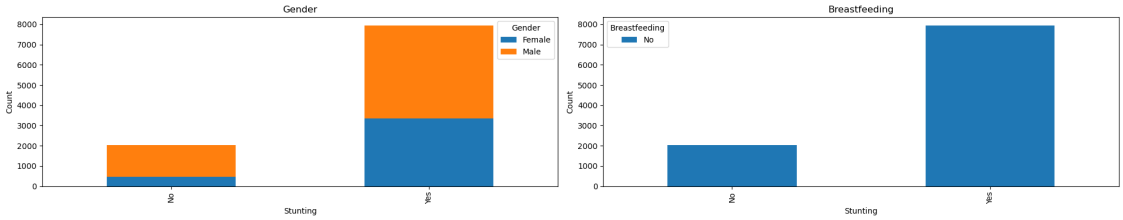
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 8 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Gender         10000 non-null  object
 1   Age            10000 non-null  int64
 2   Birth Weight   10000 non-null  float64
 3   Birth Length   10000 non-null  int64
 4   Body Weight    10000 non-null  float64
 5   Body Length    10000 non-null  float64
 6   Breastfeeding  10000 non-null  object
 7   Stunting       10000 non-null  object
dtypes: float64(3), int64(2), object(3)
memory usage: 625.1+ KB
```

[45]:
```
df.isnull().sum()
```

[45]:
```
Gender           0
Age              0
Birth Weight     0
Birth Length     0
Body Weight      0
Body Length      0
Breastfeeding    0
Stunting         0
dtype: int64
```

[46]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Distribution of categorical variables
categorical_vars = [
    'Gender', 'Breastfeeding'
]
```

```
# Buat subplots sesuai jumlah variabel kategorikal
fig, axes = plt.subplots((len(categorical_vars) + 2) // 3, 2, figsize=(20,␣
 ↪len(categorical_vars) * 2))
axes = axes.flatten()

# Plot stacked bar chart untuk setiap variabel kategorik
for i, var in enumerate(categorical_vars):
    if var in df.columns:
        counts = df.groupby(['Stunting', var]).size().unstack(fill_value=0)
        counts.plot(kind='bar', stacked=True, ax=axes[i])
        axes[i].set_title(var)
        axes[i].set_ylabel('Count')
        axes[i].set_xlabel('Stunting')
    else:
        axes[i].axis('off')

# Adjust layout
plt.tight_layout()
plt.show()
```



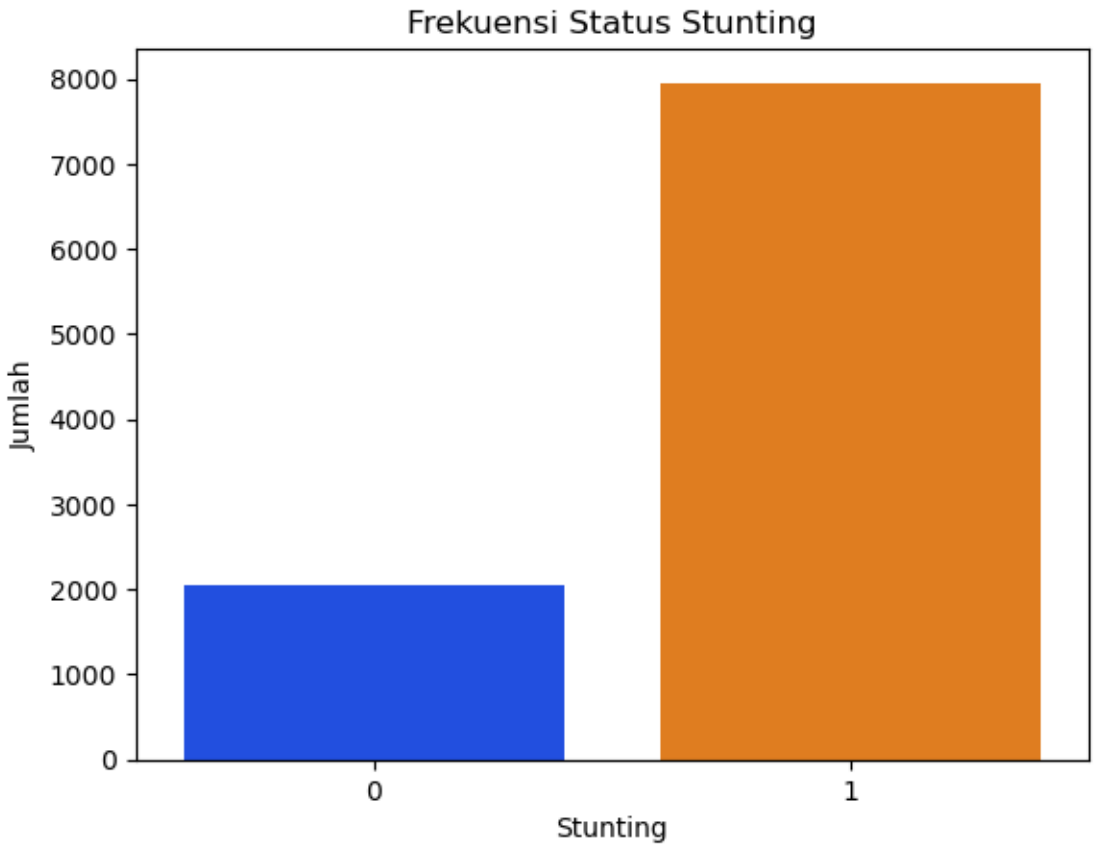
[47]:
```
df['Gender'].replace(['Male', 'Female'], [0,1], inplace=True)
df['Breastfeeding'].replace(['No', 'Yes'], [0,1], inplace=True)
df['Stunting'].replace(['No', 'Yes'], [0,1], inplace=True)
df.head(10)
```

[47]:

	Gender	Age	Birth Weight	Birth Length	Body Weight	Body Length	Breastfeeding	Stunting
0	0	17	3.0	49	10.0	72.2		
1	1	11	2.9	49	2.9	65.0		
2	0	16	2.9	49	8.5	72.2		
3	0	31	2.8	49	6.4	63.0		
4	0	15	3.1	49	10.5	49.0		
5	1	11	2.8	49	8.5	65.0		
6	0	35	2.8	49	10.5	72.2		
7	1	17	2.8	49	8.0	63.0		
8	1	10	2.7	49	8.4	73.5		
9	1	16	2.8	49	8.5	65.0		

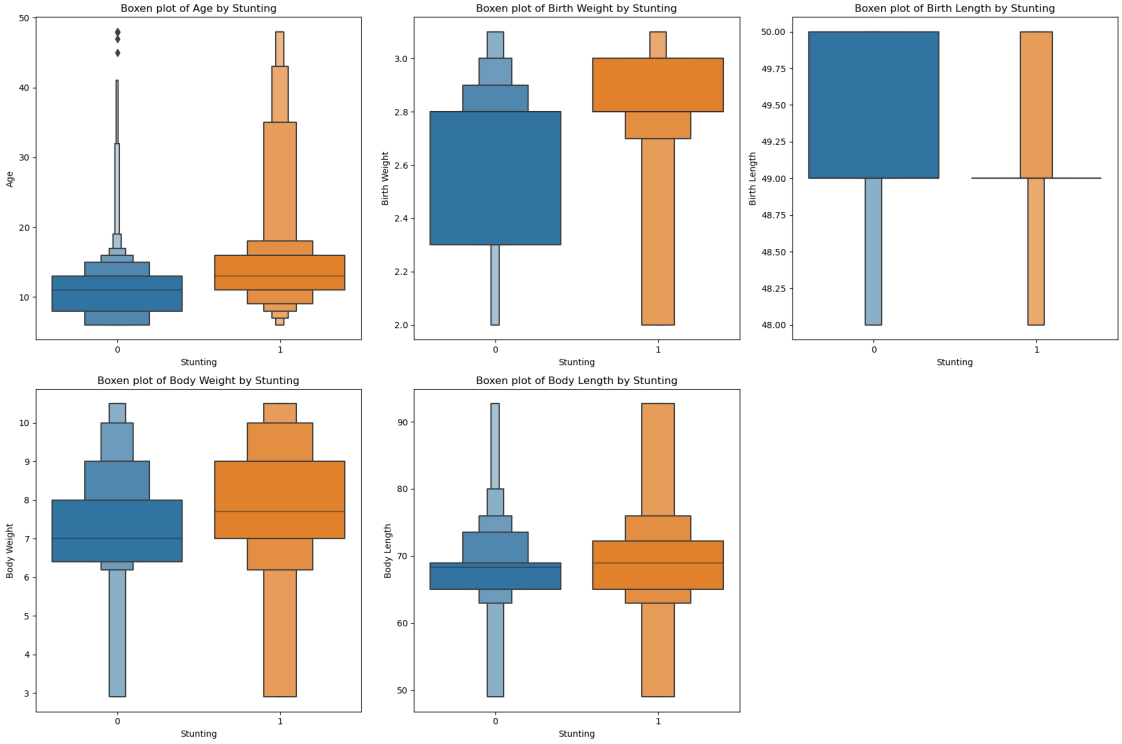
0	0	0
1	0	1
2	0	1
3	0	1
4	0	1
5	0	0
6	0	1
7	0	1
8	0	0
9	0	1

[48]:
```
#Visualisasi model

#impor library yang bisa digunakan untuk visualisasi data seperti membuat␣
 ↪grafik seperti judul, label,
#dan lain-lain.
import matplotlib.pyplot as plt

#untuk mengatur warnanya
import seaborn as sns

#ntuk menampilkan count plot, yang secara default menunjukkan frekuensi dari␣
 ↪dari variabel yang digunakan
#Menetapkan sstunting senagai variabel pada sumbu x
#Menetapkan palet warna untuk grafik
sns.countplot(x='Stunting', data=df, palette='bright')
#Menambahkan judul di atas grafik.
plt.title('Frekuensi Status Stunting')
#Menambahkan label untuk sumbu x dan y.
plt.xlabel('Stunting')
plt.ylabel('Jumlah')
#menampilkan grafiknya
plt.show()
```



[49]:
```python
# 3. Distribution of numerical columns
numeric_vars = [
    'Age',
    'Birth Weight',
    'Birth Length',
    'Body Weight',
    'Body Length'
]

# Membuat boxen plot untuk masing-masing variabel numerik
plt.figure(figsize=(18, 12))
for i, var in enumerate(numeric_vars, 1):
    plt.subplot(2, 3, i)
    sns.boxenplot(x='Stunting', y=var, data=df)
    plt.title(f'Boxen plot of {var} by Stunting')
    plt.xlabel('Stunting')
    plt.ylabel(var)
    plt.tight_layout()

plt.show()
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_2804\1506641860.py:18: UserWarning:
The figure layout has changed to tight
  plt.tight_layout()
C:\Users\ASUS\AppData\Local\Temp\ipykernel_2804\1506641860.py:18: UserWarning:
The figure layout has changed to tight
  plt.tight_layout()
C:\Users\ASUS\AppData\Local\Temp\ipykernel_2804\1506641860.py:18: UserWarning:
The figure layout has changed to tight
  plt.tight_layout()
C:\Users\ASUS\AppData\Local\Temp\ipykernel_2804\1506641860.py:18: UserWarning:
The figure layout has changed to tight
  plt.tight_layout()
```



3 RBF

[50]:
```
# Selecting the columns and converting to NumPy arrays
X = df[['Gender', 'Age', 'Birth Weight', 'Birth Length', 'Body Weight', 'Body␣
  ↪Length', 'Breastfeeding']].to_numpy()
y = df['Stunting'].to_numpy()

# Standardisasi fitur
scaler = StandardScaler()
X = scaler.fit_transform(X)
```

```
# Splitting the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# SMOTE
from imblearn.over_sampling import SMOTE

# Balancing the training data using SMOTE
smote = SMOTE(random_state=42)
X_train, y_train = smote.fit_resample(X_train, y_train)
```

[51]:
```
# Corrected RBF Network implementation
class RBFNetwork(BaseEstimator, ClassifierMixin):
    def __init__(self, k=10, gamma=None):
        self.k = k
        self.gamma = gamma

    def _rbf(self, X, centers, gamma):
        # Compute RBF (Gaussian) for each center
        G = np.exp(-gamma * np.linalg.norm(X[:, np.newaxis] - centers,␣
 ↪axis=2)**2)
        return G

    def fit(self, X, y):
        # Use KMeans to find centers
        kmeans = KMeans(n_clusters=self.k, random_state=42)
        kmeans.fit(X)
        self.centers = kmeans.cluster_centers_

        # Calculate gamma if not provided
        if self.gamma is None:
            d_max = np.max(np.linalg.norm(self.centers - self.centers[:, np.
 ↪newaxis], axis=2))
            self.gamma = 1 / (2 * (d_max / np.sqrt(2 * self.k))**2)

        # Transform X to RBF space
        G = self._rbf(X, self.centers, self.gamma)

        # Fit linear regression model in RBF space
        self.model = LinearRegression()
        self.model.fit(G, y)

        return self

    def predict(self, X):
        G = self._rbf(X, self.centers, self.gamma)
```

```
        return self.model.predict(G)
```

[52]:
```
# Train the RBF network
rbf = RBFNetwork(k=50)
rbf.fit(X_train, y_train)

# Predict the output on the test set
y_pred = rbf.predict(X_test)

# Evaluate the performance of the model using accuracy score
acc = accuracy_score(y_test, y_pred.round())
print(f"Accuracy score: {acc:.3f}")
```

```
Accuracy score: 0.798
```

4 MLP

[53]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score

# Train the MLP model
mlp = MLPClassifier(hidden_layer_sizes=(100,), max_iter=300, random_state=42)
mlp.fit(X_train, y_train)

# Predict the output on the test set
y_pred = mlp.predict(X_test)

# Evaluate the performance of the model using accuracy score
acc = accuracy_score(y_test, y_pred)
print(f"Accuracy score: {acc:.3f}")
```

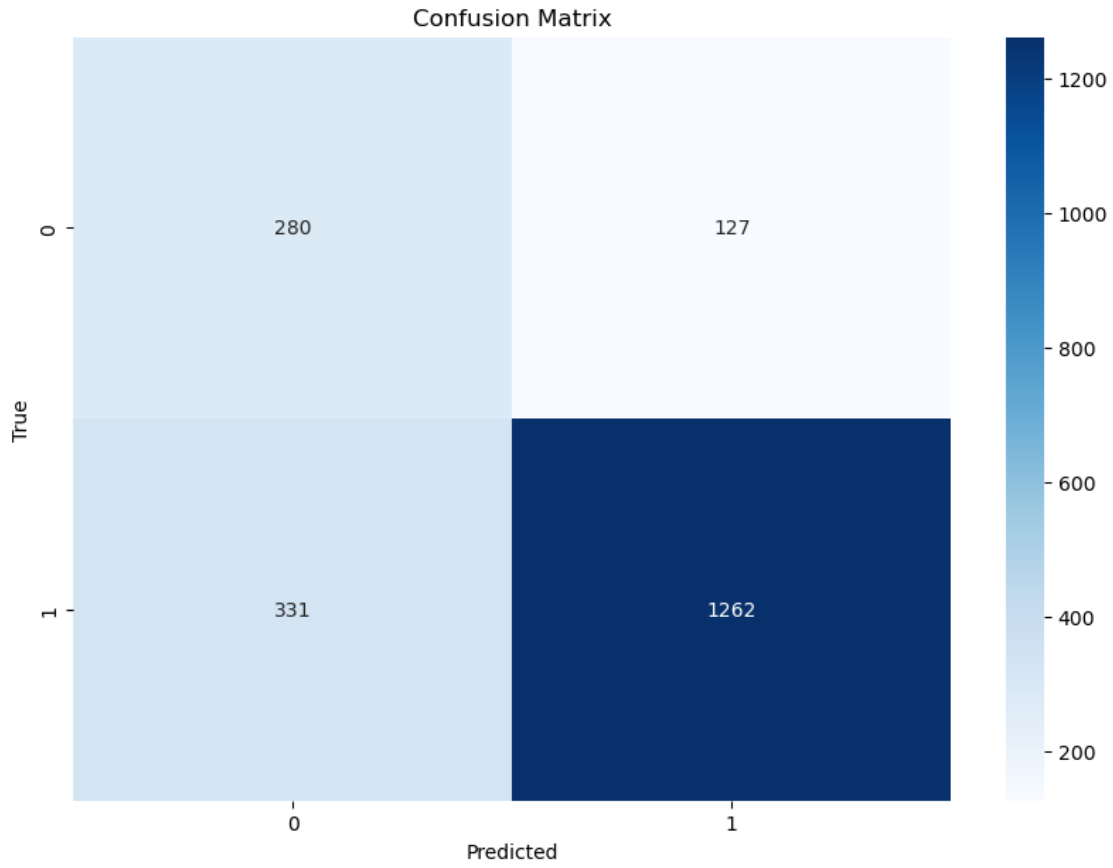
```
Accuracy score: 0.771
```

[54]:
```
import numpy as np
from hyperopt import fmin, tpe, hp, Trials, STATUS_OK
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score, confusion_matrix,␣
 ↪classification_report

# Membuat confusion matrix dan classification report
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

print("\nClassification Report:")
```

```
print(class_report)
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.46      0.69      0.55       407
           1       0.91      0.79      0.85      1593

    accuracy                           0.77      2000
   macro avg       0.68      0.74      0.70      2000
weighted avg       0.82      0.77      0.79      2000
```

[55]:
```
# Plotting confusion matrix
plt.figure(figsize=(10, 7))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()
```



5 MLP GridSearchCV

[56]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
```

[57]:
```
# Definisi parameter grid untuk GridSearchCV
param_grid = {
    'hidden_layer_sizes': [(10,), (20,), (10, 10)],
    'activation': ['tanh', 'relu'],
    'solver': ['sgd', 'adam'],
    'alpha': [0.0001, 0.05],
    'learning_rate': ['constant', 'adaptive']
}

# Inisialisasi MLPClassifier
mlp = MLPClassifier(max_iter=100)

# Inisialisasi GridSearchCV
grid_search = GridSearchCV(mlp, param_grid, n_jobs=-1, cv=3)

# Melatih model menggunakan GridSearchCV
grid_search.fit(X_train, y_train)

# Menampilkan hyperparameter terbaik
print("Best parameters found: ", grid_search.best_params_)

# Memprediksi hasil pada test set menggunakan model terbaik
best_model = grid_search.best_estimator_
y_pred = best_model.predict(X_test)

# Mengevaluasi kinerja model menggunakan accuracy score
acc = accuracy_score(y_test, y_pred)
print(f"Accuracy score: {acc:.3f}")
```

```
Best parameters found:  {'activation': 'relu', 'alpha': 0.05,
'hidden_layer_sizes': (20,), 'learning_rate': 'constant', 'solver': 'adam'}
Accuracy score: 0.777
```

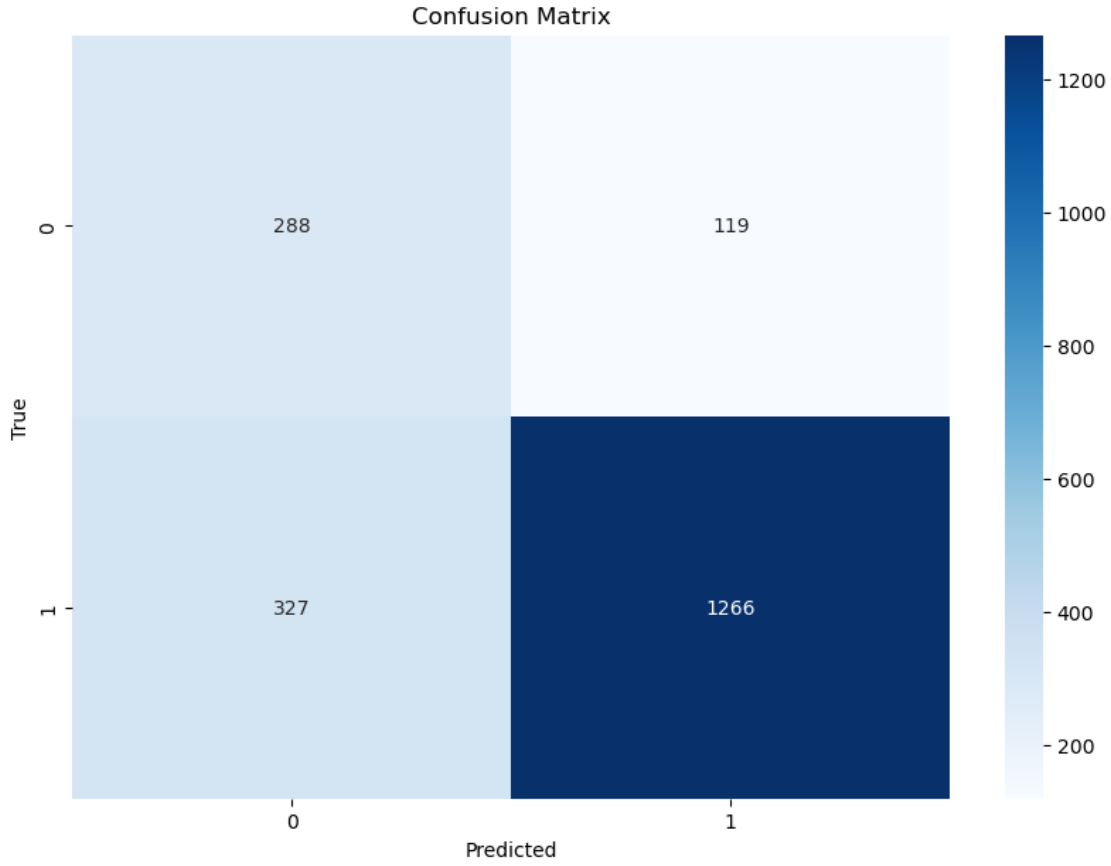
```
d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
```

```
  warnings.warn(
```

[58]:
```
import numpy as np
from hyperopt import fmin, tpe, hp, Trials, STATUS_OK
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score, confusion_matrix,␣
 ↪classification_report

# Membuat confusion matrix dan classification report
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

print("\nClassification Report:")
print(class_report)
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.47      0.71      0.56       407
           1       0.91      0.79      0.85      1593

    accuracy                           0.78      2000
   macro avg       0.69      0.75      0.71      2000
weighted avg       0.82      0.78      0.79      2000
```

[59]:
```
# Plotting confusion matrix
plt.figure(figsize=(10, 7))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()
```



6 MLP RandomizedSearchCV

[60]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, RandomizedSearchCV
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
from scipy.stats import uniform, randint
```

[61]:
```
# Definisi parameter grid untuk RandomizedSearchCV
param_dist = {
    'hidden_layer_sizes': [(10,), (20,), (10, 10)],
    'activation': ['tanh', 'relu'],
    'solver': ['sgd', 'adam'],
    'alpha': uniform(0.0001, 0.05),
    'learning_rate': ['constant', 'adaptive']
}
```

```
# Inisialisasi MLPClassifier
mlp = MLPClassifier(max_iter=100)

# Inisialisasi RandomizedSearchCV
random_search = RandomizedSearchCV(mlp, param_distributions=param_dist,␣
 ↪n_iter=100, n_jobs=-1, cv=3, random_state=42)

# Melatih model menggunakan RandomizedSearchCV
random_search.fit(X_train, y_train)

# Menampilkan hyperparameter terbaik
print("Best parameters found: ", random_search.best_params_)

# Memprediksi hasil pada test set menggunakan model terbaik
best_model = random_search.best_estimator_
y_pred = best_model.predict(X_test)

# Mengevaluasi kinerja model menggunakan accuracy score
acc = accuracy_score(y_test, y_pred)
print(f"Accuracy score: {acc:.3f}")
```

```
Best parameters found:  {'activation': 'relu', 'alpha': 0.01772844281670845,
'hidden_layer_sizes': (10, 10), 'learning_rate': 'constant', 'solver': 'adam'}
Accuracy score: 0.776
```

```
d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(
```

7 MLP Bayesian

[62]:
```
pip install bayesian-optimization
```

```
Requirement already satisfied: bayesian-optimization in d:\anaconda\lib\site-
packages (1.4.3)
Requirement already satisfied: numpy>=1.9.0 in d:\anaconda\lib\site-packages
(from bayesian-optimization) (1.24.3)
Requirement already satisfied: scipy>=1.0.0 in d:\anaconda\lib\site-packages
(from bayesian-optimization) (1.11.1)
Requirement already satisfied: scikit-learn>=0.18.0 in d:\anaconda\lib\site-
packages (from bayesian-optimization) (1.4.1.post1)
Requirement already satisfied: colorama>=0.4.6 in d:\anaconda\lib\site-packages
(from bayesian-optimization) (0.4.6)
Requirement already satisfied: joblib>=1.2.0 in d:\anaconda\lib\site-packages
(from scikit-learn>=0.18.0->bayesian-optimization) (1.2.0)
```

```
Requirement already satisfied: threadpoolctl>=2.0.0 in d:\anaconda\lib\site-
packages (from scikit-learn>=0.18.0->bayesian-optimization) (2.2.0)
Note: you may need to restart the kernel to use updated packages.
```

[63]:
```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
from bayes_opt import BayesianOptimization
```

[64]:
```python
# Fungsi untuk optimisasi Bayesian
def mlp_evaluate(hidden_layer_sizes, alpha, learning_rate_init):
    hidden_layer_sizes = int(hidden_layer_sizes)
    alpha = 10 ** alpha
    learning_rate_init = 10 ** learning_rate_init

    mlp = MLPClassifier(hidden_layer_sizes=(hidden_layer_sizes,),
                        alpha=alpha,
                        learning_rate_init=learning_rate_init,
                        max_iter=100,
                        random_state=42)
    mlp.fit(X_train, y_train)
    y_pred = mlp.predict(X_test)
    return accuracy_score(y_test, y_pred)

# Bounded region of parameter space
pbounds = {
    'hidden_layer_sizes': (10, 100),
    'alpha': (-5, 2),
    'learning_rate_init': (-5, 0)
}

# Inisialisasi Bayesian Optimization
optimizer = BayesianOptimization(
    f=mlp_evaluate,
    pbounds=pbounds,
    random_state=42,
)

# Melakukan optimisasi
optimizer.maximize(init_points=10, n_iter=30)

# Menampilkan parameter terbaik
print("Best parameters found: ", optimizer.max)
```

```
# Memprediksi hasil pada test set menggunakan model terbaik
params = optimizer.max['params']
best_mlp =␣
 ↪MLPClassifier(hidden_layer_sizes=(int(params['hidden_layer_sizes']),),
                        alpha=10 ** params['alpha'],
                        learning_rate_init=10 ** params['learning_rate_init'],
                        max_iter=100,
                        random_state=42)
best_mlp.fit(X_train, y_train)
y_pred = best_mlp.predict(X_test)

# Mengevaluasi kinerja model menggunakan accuracy score
acc = accuracy_score(y_test, y_pred)
print(f"Accuracy score: {acc:.3f}")
```

```
|   iter    |  target   |   alpha   | hidden… | learni… |
-------------------------------------------------------------
| 1         | 0.7855    | -2.378    | 95.56
| -1.34     |

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(

| 2         | 0.787     | -0.8094   | 24.04

 | -4.22     |

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(

| 3         | 0.772     | -4.593    | 87.96
| -1.994    |
| 4         | 0.807     | -0.04349  | 11.85

 | -0.1505   |

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(

| 5         | 0.788     | 0.8271    | 29.11
| -4.091    |

d:\Anaconda\Lib\site-
```

```
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(

| 6         | 0.792     | -3.716    | 37.38
| -2.376    |
| 7         | 0.7835    | -1.976    | 36.21
| -1.941    |

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(

| 8         | 0.7785    | -4.024    | 36.29
| -3.168    |

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(

| 9         | 0.781     | -1.808    | 80.67
| -4.002    |

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(

| 10        | 0.781     | -1.4      | 63.32
| -4.768    |

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(

| 11        | 0.785     | -0.5585   | 24.06
| -4.05     |
| 12        | 0.7885    | 0.6475    | 11.64
| -0.3971   |

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(
```

```
| 13        | 0.792     | -0.7501   | 24.12
| -4.295    |
| 14        | 0.7605    | 0.5199    | 12.08
| -0.3195   |
| 15        | 0.7845    | -0.1171   | 11.69
| -0.3735   |
| 16        | 0.832     | 0.638     | 11.47
 | -0.137    |

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(

| 17        | 0.7765    | -4.339    | 82.11
| -2.021    |

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(

| 18        | 0.7995    | -3.697    | 37.01
| -2.394    |
| 19        | 0.736     | 0.2999    | 11.35
| -0.0187   |
| 20        | 0.812     | 0.9048    | 11.44
| -0.3427   |

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(

| 21        | 0.783     | -2.842    | 87.5
| -4.531    |
| 22        | 0.774     | 0.7152    | 11.72
| -0.2488   |

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(

| 23        | 0.779     | -3.123    | 32.98
| -2.478    |
```

```
d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(

| 24        | 0.7775    | -0.7135   | 15.54
| -2.876    |

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(

| 25        | 0.7835    | -0.8536   | 74.77
| -4.36     |
| 26        | 0.6915    | 1.62      | 50.63
| -0.4485   |
| 27        | 0.7795    | -0.1334   | 11.8
| -0.01576  |
| 28        | 0.732     | -2.646    | 15.23
| -0.6074   |

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(

| 29        | 0.759     | -3.504    | 36.71
| -2.51     |

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(

| 30        | 0.7825    | 0.8908    | 29.15
| -3.992    |
| 31        | 0.742     | -0.05437  | 11.87
| -0.1423   |
| 32        | 0.829     | -1.313    | 29.91
| -0.3205   |
| 33        | 0.767     | -4.508    | 80.26
| -0.6374   |
| 34        | 0.7915    | -3.66     | 47.86
| -0.5825   |
| 35        | 0.8275    | -0.681    | 14.24
| -0.7779   |
```

```
| 36        | 0.7685    | 0.6931    | 78.49
| -1.724    |
```

```
d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(
```

```
| 37        | 0.7875    | -3.401    | 31.46
| -4.214    |
```

```
d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(
```

```
| 38        | 0.7485    | -4.559    | 35.88
| -1.808    |
| 39        | 0.7845    | 0.9287    | 33.77
| -2.367    |
```

```
d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(
```

```
| 40        | 0.7675    | -4.595    | 50.87
| -4.141    |
=============================================================
Best parameters found:  {'target': 0.832, 'params': {'alpha':
0.6380391579683256, 'hidden_layer_sizes': 11.469588055878903,
'learning_rate_init': -0.1369680516751206}}
Accuracy score: 0.832
```

8 MLP Hyperopt

[65]:
```
pip install hyperopt
```

```
Requirement already satisfied: hyperopt in d:\anaconda\lib\site-packages (0.2.7)
Requirement already satisfied: numpy in d:\anaconda\lib\site-packages (from
hyperopt) (1.24.3)
Requirement already satisfied: scipy in d:\anaconda\lib\site-packages (from
hyperopt) (1.11.1)
Requirement already satisfied: six in d:\anaconda\lib\site-packages (from
hyperopt) (1.16.0)
Requirement already satisfied: networkx>=2.2 in d:\anaconda\lib\site-packages
(from hyperopt) (3.1)
```

```
Requirement already satisfied: future in d:\anaconda\lib\site-packages (from
hyperopt) (0.18.3)
Requirement already satisfied: tqdm in d:\anaconda\lib\site-packages (from
hyperopt) (4.65.0)
Requirement already satisfied: cloudpickle in d:\anaconda\lib\site-packages
(from hyperopt) (2.2.1)
Requirement already satisfied: py4j in d:\anaconda\lib\site-packages (from
hyperopt) (0.10.9.7)
Requirement already satisfied: colorama in d:\anaconda\lib\site-packages (from
tqdm->hyperopt) (0.4.6)
Note: you may need to restart the kernel to use updated packages.
```

[66]:
```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
from hyperopt import hp, tpe, fmin, Trials, STATUS_OK
```

[67]:
```python
# Ruang pencarian hyperparameter
space = {
    'hidden_layer_sizes': hp.quniform('hidden_layer_sizes', 10, 100, 1),
    'activation': hp.choice('activation', ['tanh', 'relu']),
    'solver': hp.choice('solver', ['sgd', 'adam']),
    'alpha': hp.loguniform('alpha', np.log(1e-5), np.log(1e-2)),
    'learning_rate': hp.choice('learning_rate', ['constant', 'adaptive'])
}

# Fungsi objektif untuk Hyperopt
def objective(params):
    mlp = MLPClassifier(hidden_layer_sizes=(int(params['hidden_layer_sizes']),),
                        activation=params['activation'],
                        solver=params['solver'],
                        alpha=params['alpha'],
                        learning_rate=params['learning_rate'],
                        max_iter=100,
                        random_state=42)

    mlp.fit(X_train, y_train)
    y_pred = mlp.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    return {'loss': -acc, 'status': STATUS_OK}

# Optimisasi dengan Hyperopt
trials = Trials()
best = fmin(fn=objective,
```

```
            space=space,
            algo=tpe.suggest,
            max_evals=100,
            trials=trials,
            rstate=np.random.default_rng(42))

# Menampilkan parameter terbaik
print("Best parameters found: ", best)

# Memprediksi hasil pada test set menggunakan model terbaik
best_mlp = MLPClassifier(hidden_layer_sizes=(int(best['hidden_layer_sizes']),),
                         activation=['tanh', 'relu'][best['activation']],
                         solver=['sgd', 'adam'][best['solver']],
                         alpha=best['alpha'],
                         learning_rate=['constant',␣
 ↪'adaptive'][best['learning_rate']],
                         max_iter=100,
                         random_state=42)
best_mlp.fit(X_train, y_train)
y_pred = best_mlp.predict(X_test)

# Mengevaluasi kinerja model menggunakan accuracy score
acc = accuracy_score(y_test, y_pred)
print(f"Accuracy score: {acc:.3f}")
```

```
  0%|          | 0/100 [00:00<?, ?trial/s, best loss=?]  1%|          | 1/100
[00:06<10:55,  6.62s/trial, best loss: -0.771]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


  2%|          | 2/100 [00:12<09:48,  6.00s/trial, best loss: -0.7775]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


  3%|          | 3/100 [00:16<08:08,  5.03s/trial, best loss: -0.7775]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
```

```
  warnings.warn(


  4%|         | 4/100 [00:21<08:07,  5.08s/trial, best loss: -0.788]
d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


  5%|         | 5/100 [00:25<07:32,  4.77s/trial, best loss: -0.788]
d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


  6%|         | 6/100 [00:30<07:22,  4.70s/trial, best loss: -0.788]
d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


  7%|         | 7/100 [00:38<09:20,  6.03s/trial, best loss: -0.788]
d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


  8%|         | 8/100 [00:46<10:08,  6.61s/trial, best loss: -0.788]
d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


  9%|         | 9/100 [00:53<10:16,  6.77s/trial, best loss: -0.788]
d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
```

```
  warnings.warn(


 10%|          | 10/100 [01:00<09:58,  6.65s/trial, best loss: -0.788]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 11%|          | 11/100 [01:03<08:35,  5.79s/trial, best loss: -0.788]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 12%|         | 12/100 [01:09<08:20,  5.69s/trial, best loss: -0.788]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 13%|         | 13/100 [01:16<08:50,  6.10s/trial, best loss: -0.788]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 14%|         | 14/100 [01:21<08:17,  5.79s/trial, best loss: -0.788]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 15%|         | 15/100 [01:27<08:25,  5.94s/trial, best loss: -0.788]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
```

```
  warnings.warn(


 16%|         | 16/100 [01:36<09:17,  6.63s/trial, best loss: -0.788]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 17%|         | 17/100 [01:42<08:54,  6.43s/trial, best loss: -0.788]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 18%|         | 18/100 [01:46<08:04,  5.91s/trial, best loss: -0.788]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 19%|         | 19/100 [01:52<08:00,  5.93s/trial, best loss: -0.788]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 20%|         | 20/100 [01:59<08:15,  6.20s/trial, best loss: -0.788]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 21%|         | 21/100 [02:04<07:33,  5.74s/trial, best loss: -0.788]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
```

```
  warnings.warn(


 22%|         | 22/100 [02:08<07:00,  5.39s/trial, best loss: -0.788]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 23%|         | 23/100 [02:13<06:50,  5.33s/trial, best loss: -0.788]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 24%|         | 24/100 [02:18<06:35,  5.20s/trial, best loss: -0.788]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 25%|         | 25/100 [02:23<06:23,  5.12s/trial, best loss: -0.788]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 26%|         | 26/100 [02:28<06:17,  5.11s/trial, best loss: -0.7885]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 27%|         | 27/100 [02:33<06:12,  5.11s/trial, best loss: -0.7885]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
```

```
  warnings.warn(


 28%|         | 28/100 [02:39<06:16,  5.23s/trial, best loss: -0.7885]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 29%|         | 29/100 [02:43<05:47,  4.90s/trial, best loss: -0.7885]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 30%|         | 30/100 [02:48<05:46,  4.94s/trial, best loss: -0.7885]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 31%|         | 31/100 [02:55<06:16,  5.45s/trial, best loss: -0.7885]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 32%|        | 32/100 [03:01<06:15,  5.52s/trial, best loss: -0.7885]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 33%|        | 33/100 [03:05<05:41,  5.09s/trial, best loss: -0.7885]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
```

```
  warnings.warn(


 34%|        | 34/100 [03:10<05:40,  5.16s/trial, best loss: -0.7885]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 35%|        | 35/100 [03:16<06:02,  5.58s/trial, best loss: -0.7885]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 36%|        | 36/100 [03:21<05:33,  5.21s/trial, best loss: -0.7885]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 37%|        | 37/100 [03:27<05:40,  5.41s/trial, best loss: -0.7885]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 38%|        | 38/100 [03:31<05:12,  5.05s/trial, best loss: -0.7885]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 39%|        | 39/100 [03:36<05:00,  4.93s/trial, best loss: -0.7885]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
```

```
  warnings.warn(


 40%|        | 40/100 [03:41<05:04,  5.07s/trial, best loss: -0.7885]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 41%|        | 41/100 [03:45<04:41,  4.78s/trial, best loss: -0.7885]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 42%|        | 42/100 [03:52<05:06,  5.28s/trial, best loss: -0.7885]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 43%|        | 43/100 [03:59<05:31,  5.82s/trial, best loss: -0.7885]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 44%|        | 44/100 [04:03<05:02,  5.41s/trial, best loss: -0.7885]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 45%|        | 45/100 [04:09<05:10,  5.65s/trial, best loss: -0.79]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
```

```
  warnings.warn(


 46%|        | 46/100 [04:16<05:27,  6.06s/trial, best loss: -0.79]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 47%|        | 47/100 [04:22<05:21,  6.07s/trial, best loss: -0.79]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 48%|        | 48/100 [04:28<05:14,  6.05s/trial, best loss: -0.79]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 49%|        | 49/100 [04:35<05:18,  6.25s/trial, best loss: -0.79]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 50%|        | 50/100 [04:41<05:04,  6.10s/trial, best loss: -0.79]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 51%|        | 51/100 [04:45<04:32,  5.57s/trial, best loss: -0.79]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
```

```
  warnings.warn(


 52%|       | 52/100 [04:51<04:33,  5.69s/trial, best loss: -0.79]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 53%|       | 53/100 [04:59<05:05,  6.50s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 54%|       | 54/100 [05:06<04:55,  6.42s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 55%|       | 55/100 [05:13<05:01,  6.69s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 56%|       | 56/100 [05:19<04:48,  6.55s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 57%|       | 57/100 [05:27<04:54,  6.85s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
```

```
  warnings.warn(


 58%|       | 58/100 [05:34<04:50,  6.91s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 59%|       | 59/100 [05:39<04:25,  6.47s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 60%|       | 60/100 [05:48<04:39,  6.99s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 61%|       | 61/100 [05:55<04:43,  7.26s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 62%|       | 62/100 [06:02<04:27,  7.03s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 63%|       | 63/100 [06:11<04:38,  7.52s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
```

```
  warnings.warn(


 64%|       | 64/100 [06:17<04:13,  7.05s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 65%|       | 65/100 [06:22<03:53,  6.67s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 66%|       | 66/100 [06:27<03:28,  6.13s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 67%|       | 67/100 [06:34<03:25,  6.24s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 68%|       | 68/100 [06:39<03:14,  6.09s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 69%|       | 69/100 [06:44<02:53,  5.60s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
```

```
  warnings.warn(


 70%|       | 70/100 [06:52<03:06,  6.22s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 71%|       | 71/100 [06:57<02:50,  5.88s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 72%|      | 72/100 [07:02<02:41,  5.77s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 73%|      | 73/100 [07:08<02:38,  5.85s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 74%|      | 74/100 [07:14<02:28,  5.73s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 75%|      | 75/100 [07:20<02:27,  5.88s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
```

```
  warnings.warn(


 76%|      | 76/100 [07:26<02:23,  5.99s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 77%|      | 77/100 [07:33<02:24,  6.26s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 78%|      | 78/100 [07:40<02:21,  6.41s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 79%|      | 79/100 [07:47<02:17,  6.55s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 80%|      | 80/100 [07:54<02:13,  6.69s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 81%|      | 81/100 [08:00<02:03,  6.52s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
```

```
  warnings.warn(


 82%|      | 82/100 [08:05<01:52,  6.23s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 83%|      | 83/100 [08:11<01:44,  6.12s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 84%|      | 84/100 [08:17<01:36,  6.05s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 85%|      | 85/100 [08:23<01:29,  5.97s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 86%|      | 86/100 [08:30<01:26,  6.19s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 87%|      | 87/100 [08:35<01:16,  5.86s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
```

```
  warnings.warn(


 88%|      | 88/100 [08:41<01:10,  5.85s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 89%|      | 89/100 [08:47<01:05,  5.98s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 90%|      | 90/100 [08:52<00:56,  5.69s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 91%|      | 91/100 [08:58<00:51,  5.71s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 92%|      | 92/100 [09:06<00:51,  6.39s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 93%|      | 93/100 [09:12<00:44,  6.30s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
```

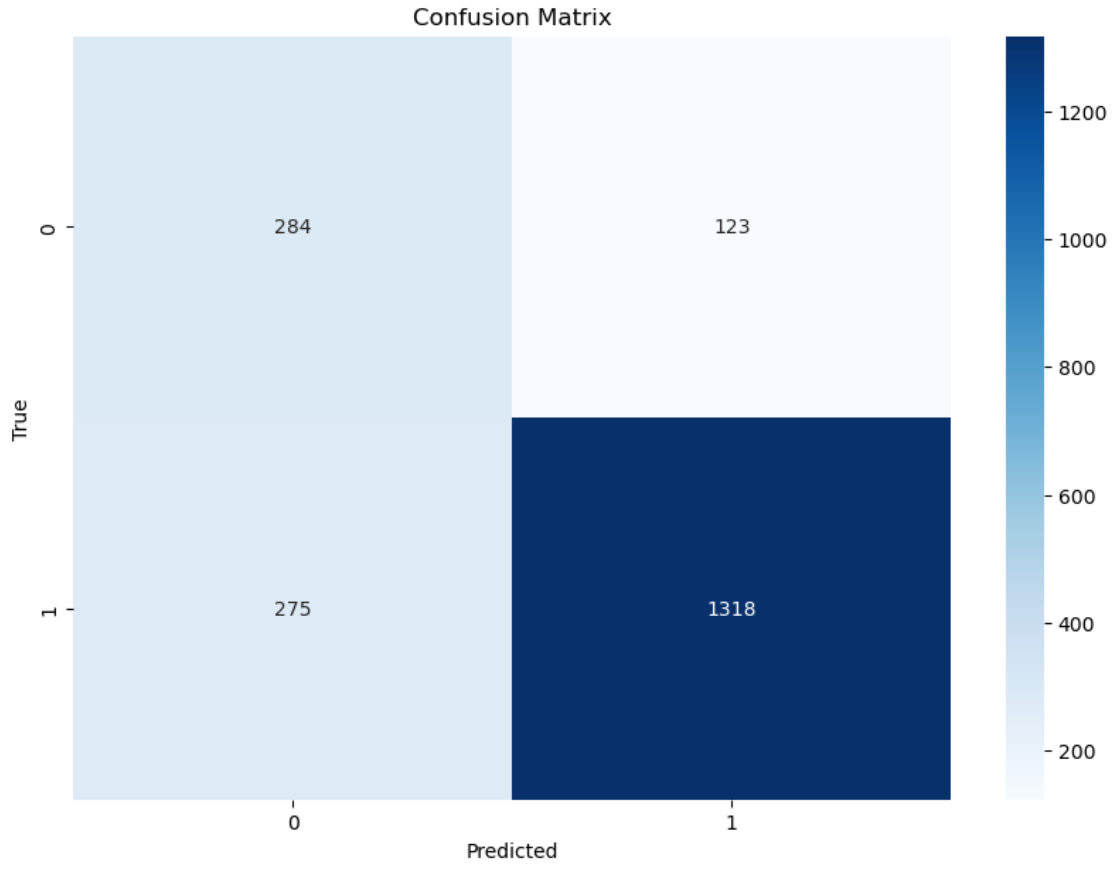
```
  warnings.warn(


 94%|      | 94/100 [09:17<00:36,  6.15s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 95%|      | 95/100 [09:22<00:28,  5.74s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 96%|      | 96/100 [09:27<00:22,  5.52s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 97%|      | 97/100 [09:32<00:15,  5.15s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 98%|      | 98/100 [09:37<00:10,  5.27s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


 99%|      | 99/100 [09:45<00:06,  6.22s/trial, best loss: -0.801]

d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
```

```
  warnings.warn(


100%|      | 100/100 [09:50<00:00,  5.91s/trial, best loss: -0.801]
d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(


Best parameters found:  {'activation': 0, 'alpha': 0.0020107901745201853,
'hidden_layer_sizes': 95.0, 'learning_rate': 1, 'solver': 1}
Accuracy score: 0.801
d:\Anaconda\Lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(
```

[68]:
```
import numpy as np
from hyperopt import fmin, tpe, hp, Trials, STATUS_OK
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score, confusion_matrix,␣
 ↪classification_report

# Membuat confusion matrix dan classification report
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

print("\nClassification Report:")
print(class_report)
```

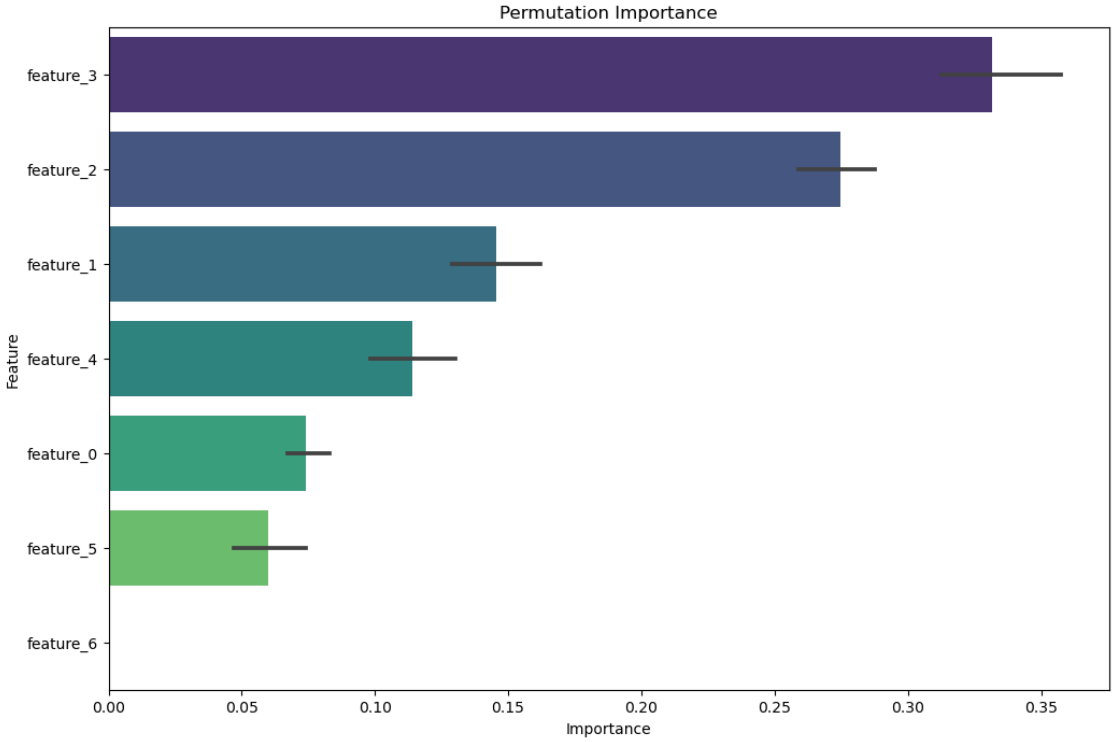
```
Classification Report:
              precision    recall  f1-score   support

           0       0.51      0.70      0.59       407
           1       0.91      0.83      0.87      1593

    accuracy                           0.80      2000
   macro avg       0.71      0.76      0.73      2000
weighted avg       0.83      0.80      0.81      2000
```

[69]:
```
# Plotting confusion matrix
plt.figure(figsize=(10, 7))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()
```



[85]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.inspection import permutation_importance

def plot_fi(model, X, y):
    if isinstance(X, pd.DataFrame):
        columns = X.columns.tolist()
    else:
        columns = [f'feature_{i}' for i in range(X.shape[1])]  # Example column␣
 ↪names for NumPy array
```

```
    perm = permutation_importance(model, X, y, n_repeats=10, random_state=0,␣
↪n_jobs=-1)
    perm2 = pd.DataFrame({'feature': columns * 10, 'importance':␣
↪perm["importances"].transpose().reshape(-1)})
    perm2["importance"] = perm2.importance / perm2.importance.sum() * 10
    urut = perm2.groupby("feature").mean().sort_values("importance",␣
↪ascending=False)

    plt.figure(figsize=(12, 8))
    sns.barplot(x='importance', y='feature', data=perm2, order=urut.index,␣
↪palette='viridis')
    plt.xlabel('Importance')
    plt.ylabel('Feature')
    plt.title('Permutation Importance')
    plt.show()

    return urut

# Example usage
plot_fi(best_mlp, X_test, y_test)
```



[85]:

feature	importance
feature_3	0.331772
feature_2	0.274671
feature_1	0.145534
feature_4	0.113909
feature_0	0.074085
feature_5	0.060029
feature_6	0.000000

[90]:
```
print(df.columns[3])
print(df.columns[2])
print(df.columns[1])
```

```
Birth Length
Birth Weight
Age
```