MEMVISUALISASIKAN DATA DENGAN *DYNAMIC PLOTTER* PADA *MACHINE LEARNING* DENGAN PENERAPAN PYTHON *CLOSURE*

Ahmad Sahidin Akbar[1], Nathanael Daniel Santoso[2], Safitri[3], Ferdy Kevin Naibaho[4], Elilya Octaviani[5]

Jurusan Sains Data, Fakultas Sains, Institut Teknologi Sumatera, Lampung Selatan, Indonesia

Email: ahmad.122450044@student.itera.ac.id, nathanael.122450059@student.itera.ac.id, safitri.122450071@student.itera.ac.id, ferdy.122450107@student.itera.ac.id, elilya.122450009@student.itera.ac.id

Abstrak

Kemajuan teknologi sudah banyak dikembangkan salah satunya *machine learning* yang harus terus dilatih sehingga banyaknya data yang perlu untuk diolah dan diinterpretasikan. Dengan memvisualisasikan sebuah data akan memudahkan pengguna untuk menyimpulkan sebuah data dan *machine learning* dapat dilatih dengan menggunakan *dynamic plotter* sebagai visualisasi data dengan penerapan python *closure.* Metode yang digunakan dengan *dynamic plotter* pada python *closure*. Penerapan python *closure* membantu *machine learning* untuk mengakses data latih dengan mudah dan efisien. *Closure* memungkinkan fungsi untuk mengakses variabel yang ditangkap bahkan setelah fungsi tersebut dipanggil di luar cakupan, mengurangi penggunaan variabel global dan membuat kode lebih sulit untuk di debug.

Kata Kunci : Data latih, *plot*, visualisasi data

PENDAHULUAN

Pada era digital ini banyak sekali teknologi-teknologi yang dibuat maupun dikembangkan. Salah satunya adalah *machine learning* yang diterapkan di kehidupan sehari-hari seperti munculnya rekomendasi barang pada pencarian di *marketplace, face ID* ataupun sidik jari dalam membuka sebuah telepon genggam

dan sebagainya. Algoritma ini menggunakan *machine learning* yang akan terus dilatih. Dengan ini tentunya membutuhkan banyak data yang diambil agar mesin dapat terus terlatih dan hasilnya menjadi lebih baik.

Dengan banyaknya data yang dihasilkan ini, banyak sekali data-data yang harus diolah dan membutuhkan sebuah interpretasi untuk menyimpulkannya. Salah satu cara untuk menginterpretasikannya dengan membuat sebuah visualisasi data tersebut dengan memilih *plot* yang sesuai dengan tipe data tersebut.

Tujuan dari judul jurnal ini untuk mengetahui apakah *machine learning* dapat dilatih dengan menggunakan *dynamic plotter* sebagai visualisasi data dengan penerapan python *closure.*

METODE

1. *Module* pada Python

   Module pada python merupakan sekumpulan kode fungsi, class dan variabel yang ada dalam file. Biasanya user harus memanggil atau mengimpor modul terlebih dahulu ke dalam kode untuk menggunakannya.

2. *Closure*

   Closure merupakan sebuah istilah dalam *programming* yang digunakan untuk mendeklarasikan suatu fungsi yang di dalamnya ada sebuah fungsi.

   Dalam Python, *closure* adalah objek fungsi yang mengingat nilai dalam lingkup terlampir, terlepas dari apakah nilai tersebut kehabisan memori. Ini adalah kumpulan data yang menyimpan fungsi beserta lingkungannya. Peta yang berisi variabel independen dari setiap fungsi (variabel digunakan secara lokal tetapi ditentukan dalam cakupan paket), ketika penutupan dibuat yang terikat oleh nilai nama atau peta referensi. Tidak seperti fungsi biasa, *closure* memungkinkan suatu fungsi mengakses variabel yang ditangkap dengan menyalin nilai atau menutup referensi bahkan jika fungsi tersebut dipanggil di luar cakupan.

   Closure di Python digunakan sebagai panggilan balik fungsi, sehingga menyediakan bentuk penyembunyian data dan membantu mengurangi penggunaan variabel global. *Closure* di Python terbukti

efektif ketika kodenya memiliki sedikit fungsi.

3. Data Latih

Penggunaan data latih pada *machine learning* agar mesin dapat mengenali pola agar hasil yang akan dihasilkan dapat memiliki galat yang kecil dan memiliki tingkat akurasi yang tinggi.

4. Visualisasi data Python

Visualisasi data adalah metode penyajian informasi data dalam bentuk grafik dan gambar yang digunakan untuk membantu memahami pola dan tren hubungan yang sulit dipahami hanya dengan melihat data dalam bentuk tabel atau numerik (Sudipa, 2023).

Pada penggunaan visualisasi data digunakan untuk melihat korelasi dari variabel yang dinyatakan menggunakan pustaka dari python. O. Wike, Claus (2019) menjelaskan bahwa visualisasi data harus ditata dan dirapikan agar mudah dipahami sehingga visualisasi tidak hanya enak dilihat namun juga jelas dan informatif.

HASIL DAN PEMBAHASAN

1. Import Library

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import
make_classification
from sklearn.model_selection
import train_test_split
from sklearn.ensemble import
RandomForestClassifier
from sklearn.metrics import
accuracy_score
```

Library numpy untuk operasi numerik. Library matplotlib.pyplot untuk plotting. Kode *make classification* dari sklearn.datasets untuk membuat data klasifikasi sintetis. Kode train_test_split dari sklearn.model_selection untuk membagi data menjadi data latih dan data uji. Kode RandomForestClassifier dari sklearn.ensemble untuk membuat model ensemble Random Forest. Kode *accuracy score* dari sklearn.metrics untuk menghitung akurasi prediksi.

2. Fungsi untuk membuat model

```
# Fungsi untuk membuat closure
model ensemble
def
create_model(X_train, y_train):
    models = []  # List untuk
menyimpan model-model individu
dalam ensemble
    def
train_and_predict(model):
```

```
model.fit(X_train,y_train)
        return model.predict
# Tambahkan model-model
individu ke dalam list
    for _ in range(3):
        model =
RandomForestClassifier(n_estima
tors=10)

models.append(train_and_predict
(model))

# Fungsi ensemble
    def ensemble_predict(X):
        predictions =
np.array([model(X) for model in
models])
        return
np.mean(predictions, axis=0)
    return ensemble_predict
```

Fungsi ini membuat model ensemble dengan menggunakan konsep closure. Membuat beberapa model Random Forest dan menyimpan fungsi predict dari setiap model. Mengembalikan fungsi *ensemble predict* yang memprediksi label rata-rata dari semua model dalam ensemble.

3. Fungsi untuk dynamic plotter

```
# Fungsi untuk dynamic plotting
def dynamic_plot(X_train,
y_train, X_test, y_test,
plot_type):
    fig, ax = plt.subplots()
    ax.set_xlabel('Number of
Estimators')
    ax.set_ylabel('Accuracy')
    ax.set_title('Dynamic Plot
of Ensemble Accuracy')

    X = np.arange(10, 100, 10)
    accuracies = []

    for n_estimators in X:
        ensemble_predict =
create_model(X_train, y_train)
        ensemble_model =
ensemble_predict(X_test)
        ensemble_model =
np.round(ensemble_model)
        accuracy =
accuracy_score(y_test,
ensemble_model)

accuracies.append(accuracy)
        if plot_type == 'line':

ax.plot(X[:len(accuracies)],
accuracies, '-o', color='b')
        elif plot_type ==
'scatter':

ax.scatter(X[:len(accuracies)],
accuracies, color='r')
        elif plot_type ==
'bar':

ax.bar(X[:len(accuracies)],
accuracies, color='g')
        else:
            print("Jenis plot
tidak valid.")
        plt.pause(0.1)
    plt.show()
```

Fungsi ini melakukan plotting dinamis dari akurasi ensemble terhadap jumlah estimator. Melakukan iterasi melalui jumlah estimator yang berbeda. Membuat model ensemble menggunakan fungsi *create model*. Menghitung akurasi prediksi ensemble dan menyimpannya. Membuat plot dengan jenis yang ditentukan oleh argumen plot *type*.

4. Membuat main program

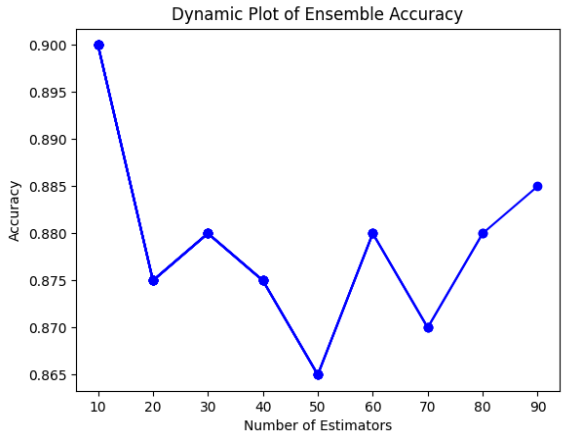
```
# Main program
if __name__ == "__main__":


# Generate data
    X, y =
make_classification(n_samples=1
000, n_features=20,
n_classes=2, random_state=42)
    X_train, X_test, y_train,
y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
# Dynamic plotting dengan jenis
plot yang berbeda
    dynamic_plot(X_train,
y_train, X_test, y_test,
'line')
    dynamic_plot(X_train,
y_train, X_test, y_test,
'scatter')
    dynamic_plot(X_train,
y_train, X_test, y_test, 'bar')
```

Memanggil fungsi *dynamic plot* untuk melakukan *plotting* dinamis dengan jenis *plot* yang berbeda (*line, scatter,* atau *bar*).

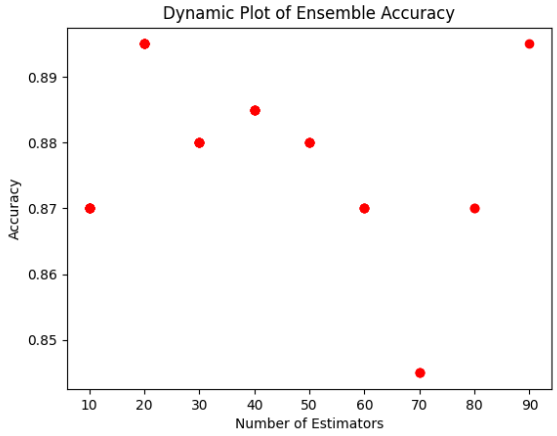
Hasil Output:

Output yang dihasilkan yaitu jendela *plot* yang menampilkan grafik dinamis dari akurasi ensemble terhadap jumlah estimator. Terdapat tiga jendela plot yang muncul secara berturut-turut, masing-masing menunjukkan jenis plot yang berbeda *(line, scatter*, atau *bar)* sesuai dengan argumen post *type* yang diberikan saat memanggil fungsi *dynamic plot*.



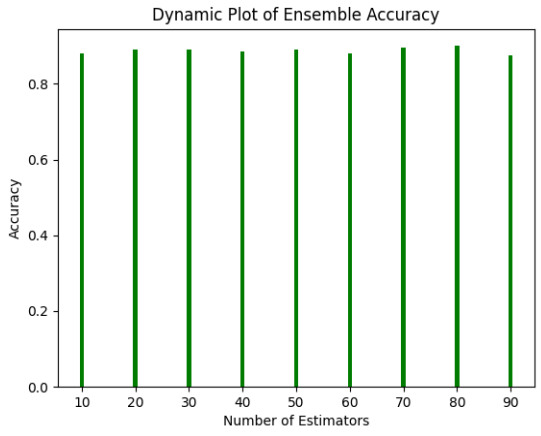
Gambar 1. Visualisasi berupa *Line Plot*

Plot ini menampilkan grafik garis yang menghubungkan titik-titik akurasi ensemble terhadap jumlah estimator. Garis ini memberikan gambaran tentang bagaimana akurasi berubah seiring dengan penambahan jumlah estimator.



Gambar 2. Visualisasi berupa *Scatter plot*

Plot ini menampilkan *scatter plot* dengan titik-titik yang merepresentasikan akurasi ensemble terhadap jumlah estimator. Setiap titik mewakili akurasi ensemble pada suatu titik waktu tertentu.



Gambar 3. Visualisasi berupa *Bar Plot*

Plot ini menampilkan *bar plot* dengan batang yang merepresentasikan akurasi ensemble pada setiap jumlah estimator. Setiap batang mewakili akurasi ensemble pada interval tertentu dari jumlah estimator.

Setiap *plot* akan berhenti sejenak (0.1 detik) setiap kali ada pembaruan data, sehingga memberikan efek dinamis saat melihat perkembangan akurasi ensemble seiring waktu.

KESIMPULAN

Berdasarkan penelitian yang dilakukan, dapat disimpulkan bahwa *machine learning* dapat dilatih dengan menggunakan *dynamic plotter* sebagai visualisasi data dengan penerapan python *closure*. Penggunaan *dynamic plotter* sebagai visualisasi data membantu machine learning untuk mengenali pola data dan meningkatkan akurasi hasil. Penerapan python *closure* membantu *machine learning* untuk mengakses data latih dengan mudah dan efisien. *Closure* memungkinkan fungsi untuk mengakses variabel yang ditangkap bahkan setelah fungsi tersebut dipanggil di luar cakupan, mengurangi penggunaan variabel global dan membuat kode lebih sulit untuk di debug.

DAFTAR PUSTAKA

Jurnal Publikasi Teknik Informatika (JUPTI). (2023, Mei). *Penggunaan Bahasa Pemrograman Python Untuk Memvisualisasikan Data Peluang Selamat Dari Kecelakaan Titanic*, *2*(2), 71. https://doi.org/10.55606/jupti.v2i2.1735

Miftah, S. (2021, May 17). *Library Python Kenali Perbedaan Module Package dan Library Pada Python*. DQLab. Retrieved April 11, 2024, from https://dqlab.id/library-python-kenali-perbedaan-module-package-dan-library-pada-python

Prayogo, N. A. (n.d.). *Python Closure | Dasar Pemrograman Python*. Dasar Pemrograman Python. Retrieved April 11, 2024, from https://dasarpemrogramanpython.novalagung.com/basic/closure

Sudipa, I. G. I., Sarasvananda, I. B. G., Prayitno, H., Putra, I. N. T. A., Darmawan, R., & WP, D. A. (2023). Teknik Visualisasi Data. PT. Sonpedia Publishing Indonesia