Tugas 3: Tugas Praktikum Mandiri 5 – Machine Learning

Alia Maisyarah 1 - 0110224095 [1*]

[1] Teknik Informatika, STT Terpadu Nurul Fikri, Depok

*E-mail:40110224095@student.nurulfikri.ac.id – **email mahasiswa 1**

1. Latihan Mandiri 5

1. # menghubungkan colab dengan google drive
   from google.colab import drive
   drive.mount('/content/gdrive')

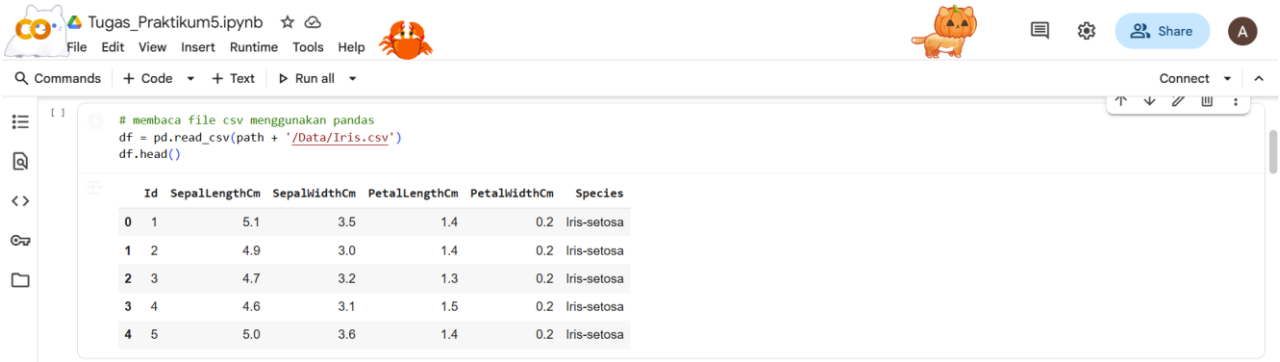
   (Perintah ini digunakan supaya file di Google Drive bisa dipakai di Colab — misalnya buat membaca dataset (day.csv, hour.csv), menyimpan hasil, atau memuat model.)

2. # memanggil data set lewat gdrive
   path = "/content/gdrive/MyDrive/Praktikum/Praktikum4"

   (Kode ini digunakan untuk menunjukkan lokasi dataset di Google Drive, supaya nanti ketika kamu memanggil data (misalnya dengan pd.read_csv(path + '/day.csv')), Python tahu harus mencari file-nya di mana.)

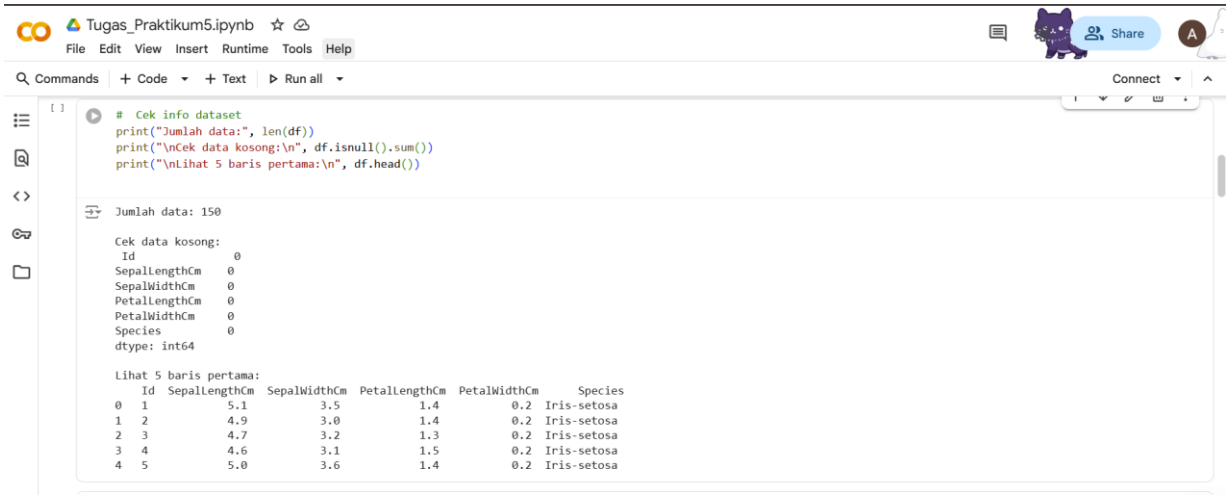
3. Import library
   import pandas as pd
   from sklearn.model_selection import train_test_split
   from sklearn.tree import DecisionTreeClassifier, plot_tree
   from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
   import matplotlib.pyplot as plt

   (Kode ini berfungsi untuk menyiapkan semua pustaka yang diperlukan dalam pembuatan, pelatihan, evaluasi, dan visualisasi model *Decision Tree* menggunakan dataset Iris.)



4. membaca file csv menggunakan pandas
df = pd.read_csv(path + '/Data/Iris.csv')
df.head()

(Kode ini digunakan untuk membaca dataset *Iris.csv* dari Google Drive menggunakan pandas, lalu menampilkan 5 baris pertama untuk memastikan data berhasil dimuat dengan benar.)



5. Cek info dataset
print("Jumlah data:", len(df))
print("\nCek data kosong:\n", df.isnull().sum())
print("\nLihat 5 baris pertama:\n", df.head())

(Kode ini digunakan untuk melakukan pemeriksaan awal pada dataset Iris, meliputi jumlah data, pengecekan nilai kosong, dan menampilkan 5 data pertama untuk memastikan data bersih dan siap digunakan

6. Pisahkan fitur (X) dan target (y)
   X = df[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']]
   y = df['Species']

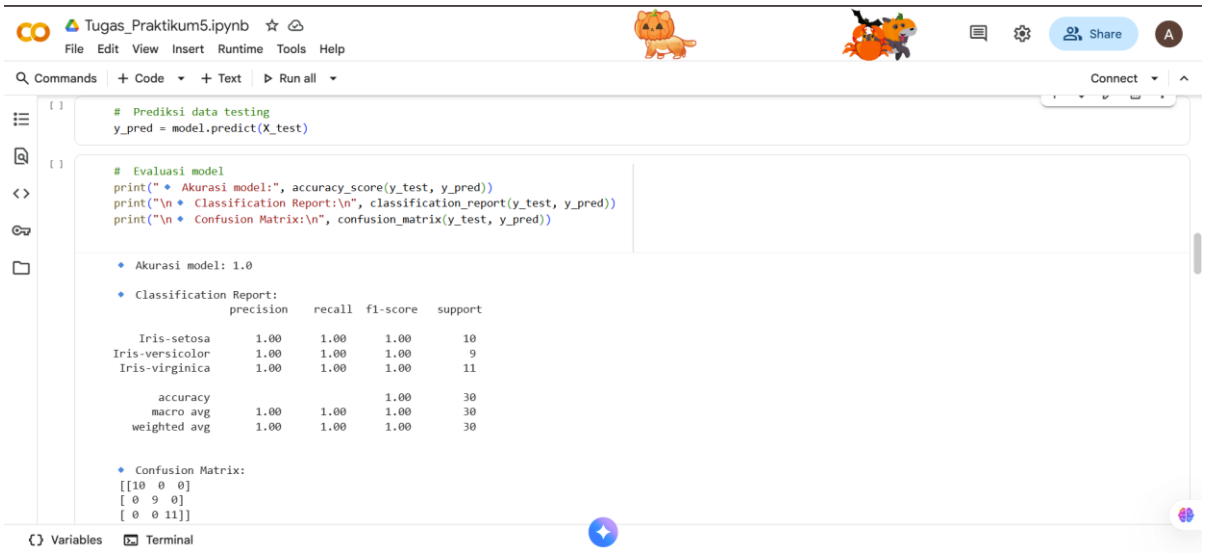
   (Pada tahap ini, dataset dipisahkan menjadi fitur (X) yang berisi data karakteristik bunga, dan target (y) yang berisi jenis spesies bunga. Pemisahan ini dilakukan agar model dapat mempelajari hubungan antara fitur dan target.)

7. Bagi data menjadi 80% training dan 20% testing
   X_train, X_test, y_train, y_test = train_test_split(
   X, y, test_size=0.2, random_state=42
)
   print("Jumlah data training:", len(X_train))
   print("Jumlah data testing:", len(X_test))

   (Dataset dibagi menjadi dua bagian: 80% untuk data pelatihan (training) dan 20% untuk data pengujian (testing). Tujuannya agar model dapat belajar dari data latih dan diuji menggunakan data yang belum pernah dilihat untuk mengukur akurasi model.)

8. Buat model Decision Tree
   model = DecisionTreeClassifier(random_state=42)
   model.fit(X_train, y_train)

   (Pada tahap ini dibuat model Decision Tree yang dilatih menggunakan data training. Model belajar mengenali pola dari fitur-fitur (panjang dan lebar sepal/petal) untuk memprediksi spesies bunga.)

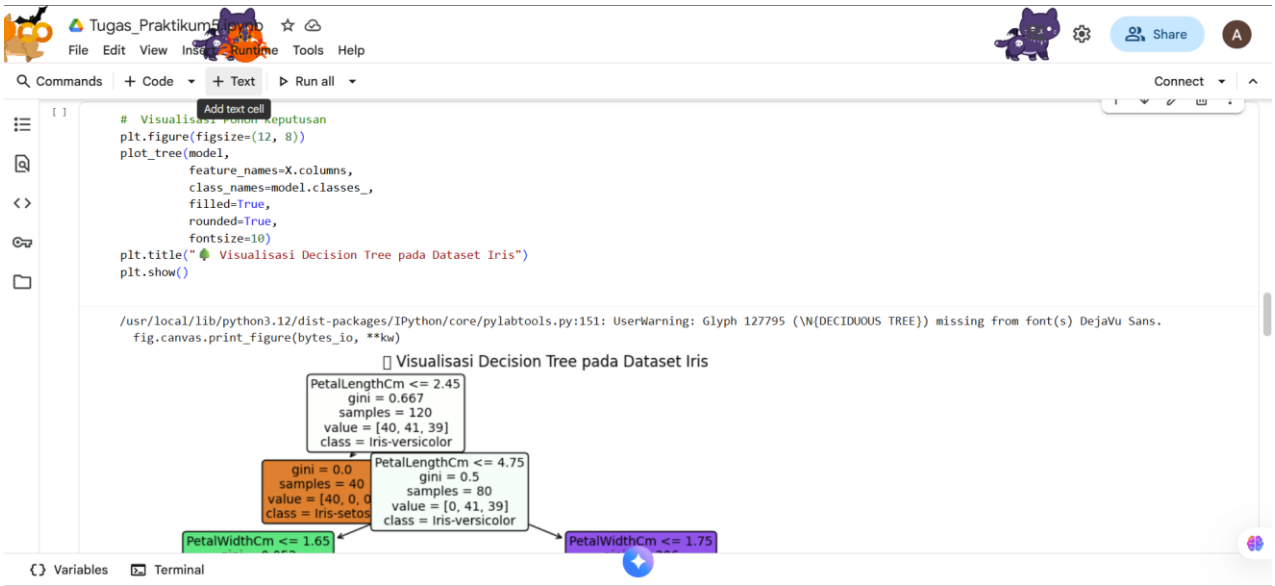


9. Prediksi data testing
   y_pred = model.predict(X_test)

   (Model digunakan untuk memprediksi data testing agar dapat diketahui seberapa baik model mengenali pola dan menentukan spesies bunga yang benar)

10. Evaluasi model
    print(" 🔹 Akurasi model:", accuracy_score(y_test, y_pred))
    print("\n 🔹 Classification Report:\n", classification_report(y_test, y_pred))
    print("\n 🔹 Confusion Matrix:\n", confusion_matrix(y_test, y_pred))

    (Tahap evaluasi digunakan untuk mengukur performa model dengan melihat tingkat akurasi, laporan klasifikasi, dan confusion matrix agar diketahui seberapa tepat model dalam memprediksi spesies bunga.)



11. Visualisasi Pohon Keputusan

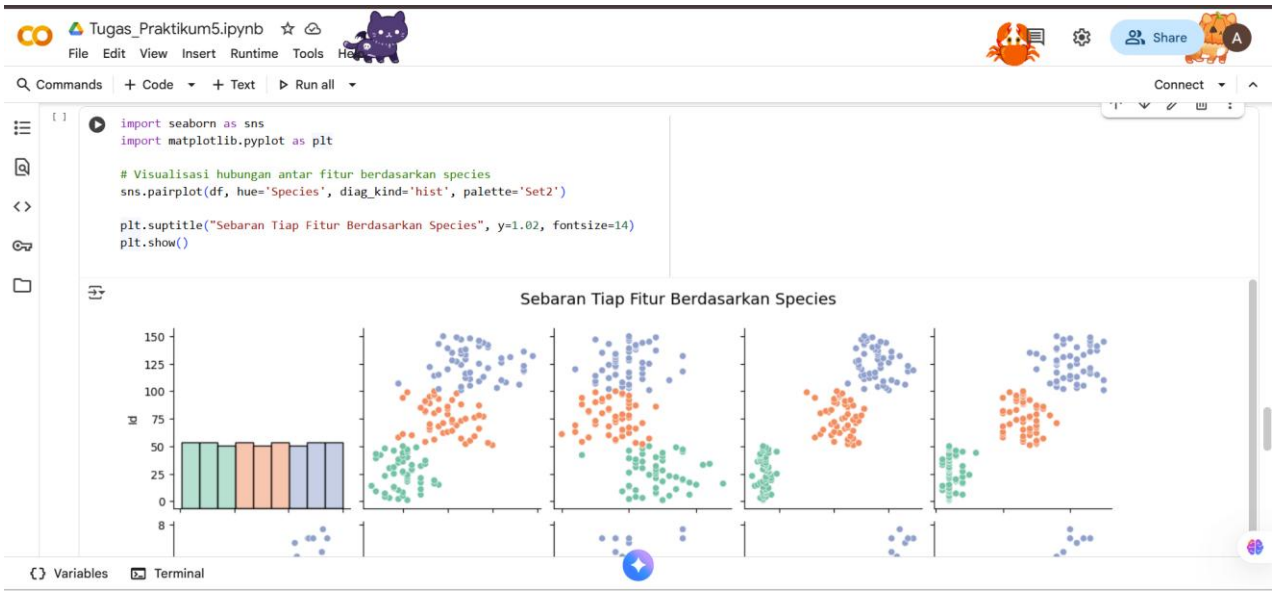
```
plt.figure(figsize=(12, 8))
plot_tree(model,
          feature_names=X.columns,
          class_names=model.classes_,
          filled=True,
          rounded=True,
          fontsize=10)
plt.title(" 🌳 Visualisasi Decision Tree pada Dataset Iris")
plt.show()
```

(Visualisasi Decision Tree menampilkan bagaimana model mengambil keputusan berdasarkan fitur-fitur (seperti panjang dan lebar kelopak) untuk mengklasifikasikan bunga ke dalam tiga jenis spesies: *Setosa*, *Versicolor*, dan *Virginica*.)

12. Uji prediksi pada satu data testing

```
sample = X_test.iloc[0:1]
print("Data uji:\n", sample)
print("\nHasil prediksi model:", model.predict(sample))
print("Label sebenarnya:", y_test.iloc[0])
```

(Bagian ini menguji satu data secara manual untuk memastikan kemampuan model dalam memprediksi spesies bunga. Jika hasil prediksi sama dengan label sebenarnya, maka model sudah mampu mengenali pola data dengan baik.)



13. import seaborn as sns
import matplotlib.pyplot as plt

# Visualisasi hubungan antar fitur berdasarkan species
sns.pairplot(df, hue='Species', diag_kind='hist', palette='Set2')

plt.suptitle("Sebaran Tiap Fitur Berdasarkan Species", y=1.02, fontsize=14)
plt.show()

(Kode ini digunakan untuk menampilkan visualisasi hubungan antar fitur bunga (panjang sepal, lebar sepal, panjang petal, dan lebar petal) berdasarkan masing-masing spesies. Grafik ini membantu melihat perbedaan pola dan sebaran antar jenis bunga Iris.)

Kesimpulan : Model *Decision Tree* dapat digunakan untuk mengklasifikasikan spesies bunga dengan akurasi tinggi. Visualisasi data membantu memahami pola hubungan antar fitur, dan hasil prediksi menunjukkan bahwa model mampu belajar dengan efektif dari dataset Iris.

LINK GITHUB UPLOAD TUGAS :

https://github.com/AliaMaisyarah14/Praktikum02ML.git