

Laporan Tugas Praktikum 05



Muhammad Bintang Gunawan - 0110224003

Teknik Informatika, STT Terpadu Nurul Fikri, Depok

0110224003@student.nurulfikri.ac.id

Abstract

Pada praktikum kelima ini, pembahasan berfokus pada penerapan algoritma *Decision Tree* untuk klasifikasi data stunting. Prosesnya meliputi pembersihan dan pengolahan data, pembagian data menjadi training dan testing, hingga evaluasi model menggunakan metrik akurasi, confusion matrix, dan classification report.

1. Tugas Mandiri

Berikut adalah kode berserta penjelasan dari praktikum Tugas Praktikum Mandiri.

```
▶ X = data.iloc[:, 1:-1]
y = data.iloc[:, -1]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

model = DecisionTreeClassifier(random_state=42)

model.fit(X_train, y_train)

print("\nModel Decision Tree berhasil dibuat dan dilatih.")

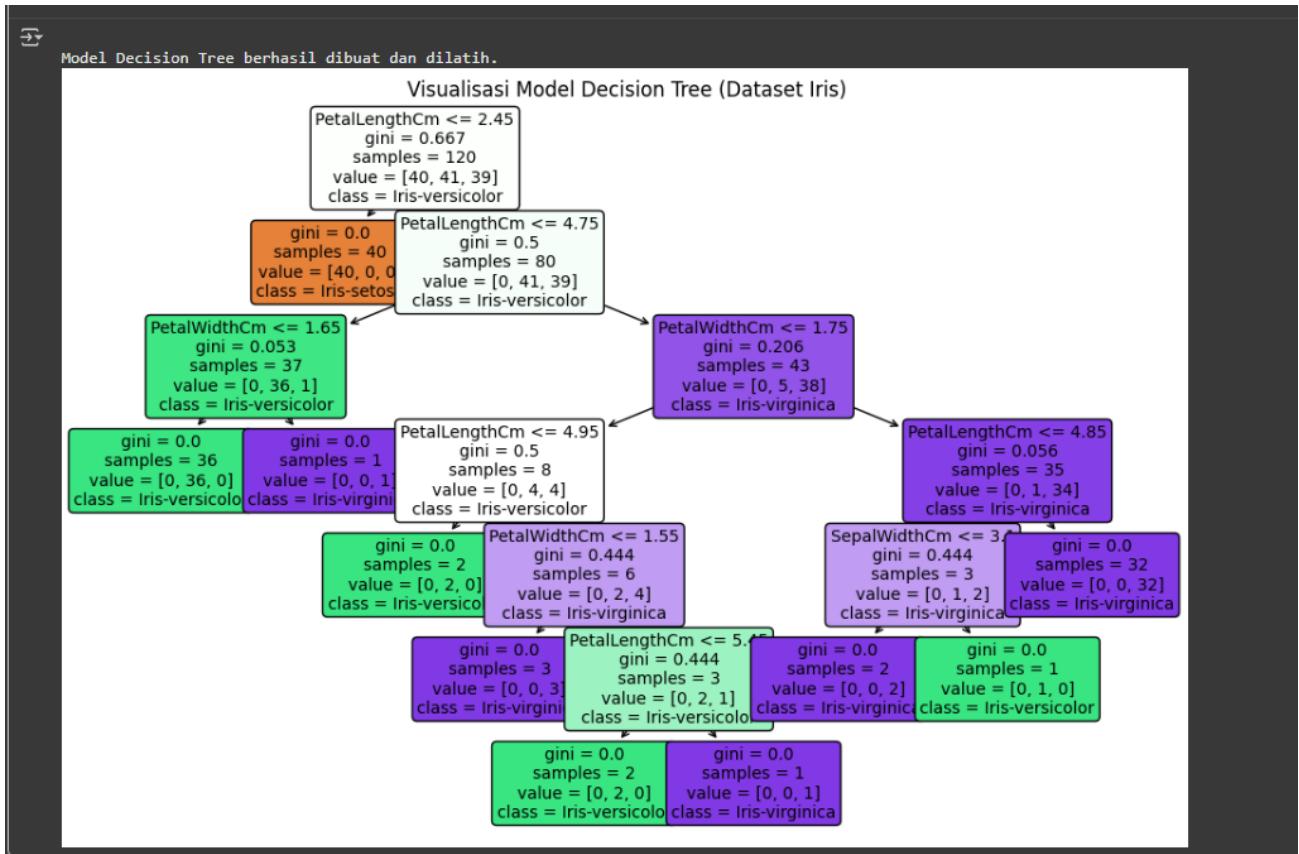
plt.figure(figsize=(12,8))
plot_tree(
    model,
    feature_names=X.columns,
    class_names=model.classes_,
    filled=True,
    rounded=True,
    fontsize=10
)
plt.title("Visualisasi Model Decision Tree (Dataset Iris)")
plt.show()
```

Penjelasan

- Bagian awal memisahkan data menjadi fitur (X) dan label target (y), lalu membagiinya menjadi data latih dan data uji menggunakan `train_test_split` (80% data untuk pelatihan, 20% untuk pengujian).
- Model Decision Tree dibuat dan dilatih dengan `model.fit(X_train, y_train)`.
- Setelah pelatihan selesai, model berhasil dibuat dan pesan “Model Decision Tree berhasil dibuat dan dilatih.” ditampilkan.

- Bagian terakhir menampilkan **visualisasi pohon keputusan** menggunakan plot_tree.

Output



Penjelasan

Pada output grafik, ditunjukkan struktur pohon keputusan berdasarkan fitur seperti *PetalLengthCm*, *PetalWidthCm*, dan lainnya.

- Setiap kotak (node) menunjukkan kondisi pemisahan data berdasarkan nilai fitur.
- Warna menunjukkan kelas hasil prediksi (setosa, versicolor, atau virginica).
- Nilai gini menunjukkan tingkat kemurnian node (semakin kecil nilainya, semakin homogen datanya).

```

y_pred = model.predict(X_test)

akurasi = accuracy_score(y_test, y_pred)
print("\nAkurasi Model:", akurasi)

print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))

print("\nClassification Report:")
print(classification_report(y_test, y_pred))

hasil = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
print("\nPerbandingan Data Aktual dan Prediksi:")
print(hasil.head())

```

Akurasi Model: 1.0

Confusion Matrix:

[[10 0 0]
[0 9 0]
[0 0 11]]

Classification Report:

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	10
Iris-versicolor	1.00	1.00	1.00	9
Iris-virginica	1.00	1.00	1.00	11
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30

Perbandingan Data Aktual dan Prediksi:

	Actual	Predicted
73	Iris-versicolor	Iris-versicolor
18	Iris-setosa	Iris-setosa
118	Iris-virginica	Iris-virginica
78	Iris-versicolor	Iris-versicolor
76	Iris-versicolor	Iris-versicolor

Penjelasan

- `y_pred = model.predict(X_test)`
Model digunakan untuk memprediksi data uji (`X_test`).
- `accuracy_score` menghitung tingkat ketepatan prediksi model terhadap data sebenarnya.
Hasil: Akurasi 1.0, artinya semua prediksi benar (100% akurat).
- `confusion_matrix` menampilkan tabel perbandingan antara data aktual dan hasil prediksi.
Hasil [[10 0 0], [0 9 0], [0 0 11]] menunjukkan tidak ada kesalahan klasifikasi antar kelas (setiap kelas diprediksi sempurna).

- `classification_report` memberikan metrik evaluasi detail seperti *precision*, *recall*, dan *f1-score* untuk setiap kelas.
Semua bernilai 1.00, yang berarti model sangat baik dalam mengklasifikasi setiap kelas (Iris-setosa, Iris-versicolor, dan Iris-virginica).
- Bagian terakhir (hasil) menampilkan perbandingan data aktual dan hasil prediksi dalam bentuk tabel, memperlihatkan bahwa semua hasil prediksi sesuai dengan label sebenarnya.

Kesimpulan hasil implementasi algoritma

Pada praktikum ini, algoritma *Decision Tree Classifier* berhasil diterapkan untuk melakukan klasifikasi pada dataset Iris. Model mampu mempelajari pola dari fitur bunga seperti panjang dan lebar kelopak serta sepal dengan sangat baik, menghasilkan akurasi 100%. Hasil evaluasi melalui *confusion matrix* dan *classification report* menunjukkan bahwa semua kelas (Iris-setosa, Iris-versicolor, dan Iris-virginica) terprediksi dengan benar tanpa kesalahan. Hal ini membuktikan bahwa algoritma *Decision Tree* efektif digunakan untuk kasus klasifikasi dengan data yang terstruktur dan memiliki pola yang jelas.

Link GitHub:

https://github.com/XiaoMao15/ML_Praktikum-dan-Tugas.git