

Tugas 5 Praktikum Mandiri

SYAHRI GHIFARI MAULIDI 0110222217

¹Teknik Informatika, STT Terpadu Nurul Fikri, Depok

1. Import Library dan Membaca Dataset

```
import pandas as pd
from sklearn import tree
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
df = pd.read_csv('../Data/Iris.csv')
print(df.head())
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

Bagian ini digunakan untuk memuat seluruh library yang dibutuhkan dalam analisis.

- pandas digunakan untuk membaca dan mengelola data.
- matplotlib.pyplot digunakan untuk menampilkan visualisasi pohon keputusan.
- sklearn.model_selection berfungsi untuk membagi dataset menjadi data latih dan data uji.
- DecisionTreeClassifier adalah algoritma yang digunakan untuk membuat model klasifikasi.
- accuracy_score, classification_report, dan confusion_matrix digunakan untuk mengevaluasi hasil prediksi model.

Selanjutnya, dataset Iris dibaca menggunakan `pd.read_csv()`. Fungsi `df.head()` menampilkan 5 baris pertama dataset agar dapat memastikan bahwa data telah berhasil dimuat. Dataset berisi kolom Id, SepalLengthCm, SepalWidthCm, PetalLengthCm, PetalWidthCm, dan Species yang menjadi target klasifikasi.

```
X = df[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']]
y = df['Species']

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

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Id              150 non-null   int64
1   SepalLengthCm   150 non-null   float64
2   SepalWidthCm    150 non-null   float64
3   PetalLengthCm   150 non-null   float64
4   PetalWidthCm    150 non-null   float64
5   Species         150 non-null   object
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

1.1 Pemisahan Data dan Informasi Dataset

Pada tahap ini, dataset dipisahkan menjadi fitur (X) dan target (y). Fitur merupakan empat atribut pengukuran bunga iris, sedangkan target adalah jenis spesies bunga. Data kemudian dibagi menjadi 80% untuk pelatihan (training) dan 20% untuk pengujian (testing) menggunakan fungsi `train_test_split`. Argumen `random_state=42` digunakan agar pembagian data selalu konsisten ketika dijalankan ulang. Fungsi `df.info()` menampilkan tipe data dan jumlah entri dari setiap kolom, menunjukkan bahwa dataset terdiri dari 150 baris tanpa nilai yang hilang.

1.2 Pelatihan Model Decision Tree

```
model = DecisionTreeClassifier(random_state=42)
model.fit(X_train, y_train)
```

▼ DecisionTreeClassifier ⓘ ?

► Parameters

Bagian ini digunakan untuk membuat dan melatih model Decision Tree. Algoritma Decision Tree bekerja dengan memecah dataset berdasarkan fitur yang paling berpengaruh terhadap target, hingga mencapai kondisi klasifikasi terbaik.

Parameter `random_state=42` menjaga agar struktur pohon yang terbentuk selalu sama setiap kali kode dijalankan. Setelah perintah `fit()`, model sudah “belajar” dari pola yang terdapat pada data latih.

2. Evaluasi Model

```
y_pred = dt_model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print("Hasil Evaluasi Model")
print('Akurasi:', round(accuracy_score(y_test, y_pred)*100, 2), "%")

print("Classification Report")
print(classification_report(y_test, y_pred))
```

Hasil Evaluasi Model
Akurasi: 96.67 %
Classification Report

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	10
Iris-versicolor	1.00	0.90	0.95	10
Iris-virginica	0.91	1.00	0.95	10
accuracy			0.97	30
macro avg	0.97	0.97	0.97	30
weighted avg	0.97	0.97	0.97	30

Tahapan ini digunakan untuk mengukur performa model terhadap data uji. Fungsi `model.predict()` menghasilkan prediksi kelas berdasarkan data yang belum pernah dilihat oleh model (data testing). Nilai akurasi dihitung untuk mengetahui persentase prediksi yang benar. Kemudian, `classification_report()` menampilkan precision, recall, dan f1-score untuk setiap kelas (Iris-setosa, Iris-versicolor, dan Iris-virginica).

Hasil evaluasi Adalah 96.67 model mampu mengklasifikasikan bunga dengan benar, Nilai precision dan recall yang tinggi menandakan bahwa model memiliki performa yang sangat baik dan mampu mengenali semua kelas dengan akurat.

2.1 Visualisasi Pohon Keputusan

```
plt.figure(figsize=(15, 10))
plot_tree(
    dt_model,
    feature_names=X.columns,
    class_names=dt_model.classes_,
    filled=True,
    rounded=True,
    fontsize=10
)
plt.title("Decision Tree Klasifikasi Iris (max_depth=4)")
plt.show()
```

Kode ini menampilkan visualisasi struktur pohon keputusan. Setiap kotak pada pohon menunjukkan node keputusan yang berisi kondisi pemisahan data berdasarkan nilai fitur tertentu. [PetalLengthCm <= 2.45] yang memisahkan data menjadi dua cabang berdasarkan panjang kelopak bunga. Bagian berwarna menandakan kelas prediksi utama di setiap node. Visualisasi ini membantu memahami bagaimana model membuat keputusan klasifikasi untuk setiap spesies bunga Iris.

2.2 Analisis Pentingnya Fitur

```
importance = pd.Series(
    dt_model.feature_importances_,
    index=X.columns
).sort_values(ascending=False)

print("Feature Importance")
print(importance)

plt.figure(figsize=(8, 5))
sns.barplot(x=importance.values, y=importance.index)
plt.title("Pentingnya Fitur (Decision Tree)")
plt.xlabel("Importance Score")
plt.ylabel("Fitur")
plt.show()
```

```
Feature Importance
PetalLengthCm    0.579077
PetalWidthCm     0.420923
SepalWidthCm     0.000000
SepalLengthCm    0.000000
dtype: float64
```

Bagian ini digunakan untuk mengetahui fitur mana yang paling berpengaruh terhadap keputusan model. Nilai feature importance dihitung berdasarkan seberapa besar peran suatu fitur dalam membagi data dengan benar pada setiap node pohon. Artinya, model Decision Tree lebih banyak menggunakan panjang dan lebar kelopak (PetalLengthCm, PetalWidthCm) untuk menentukan jenis bunga, sedangkan ukuran sepal (kelopak luar) tidak terlalu berpengaruh terhadap hasil klasifikasi.