Tugas 1: Praktikum Mandiri 5

Aisyah Hanani - 0110222286

Teknik Informatika, STT Terpadu Nurul Fikri, Depok

*E-mail: aisyahhanani82@gmail.com

Abstract. Penelitian ini bertujuan untuk mengklasifikasikan spesies bunga Iris menggunakan algoritma Decision Tree berdasarkan atribut sepal dan petal, yaitu Sepal Length, Sepal Width, Petal Length, dan Petal Width. Dataset yang digunakan merupakan dataset Iris yang diperoleh dari e-learning Elena dan telah melalui tahap pemisahan data menjadi data latih dan data uji dengan perbandingan 80% untuk data latih dan 20% untuk data uji. Model Decision Tree dibangun menggunakan bahasa pemrograman Python dan library scikit-learn. Evaluasi performa model dilakukan menggunakan metrik akurasi, confusion matrix, dan classification report. Hasil pengujian menunjukkan bahwa model Decision Tree mampu mengklasifikasikan spesies Iris dengan tingkat akurasi yang tinggi, sehingga dapat disimpulkan bahwa algoritma Decision Tree efektif digunakan untuk permasalahan klasifikasi pada dataset Iris.

1. Menyambungkan Google Colab dengan Drive

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Baris ini mengimpor modul drive dari library google colab, yang berisi fungsi untuk mengakses Google drive dari colab, kemudian memasang google drive ke direktori di Colab.

2. Import Library dan Membaca Data

Tahap pertama yaitu mengimpor berbagai library yang dibutuhkan. Library seperti pandas digunakan untuk membaca dan mengelola data, numpy untuk operasi numerik, matplotlib dan seaborn untuk visualisasi data, serta sklearn untuk membangun dan mengevaluasi model machine learning.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import matplotlib.pyplot as plt
```

2.1 Import data

2.2 Membaca Data

Dataset dibaca menggunakan pandas.read_csv(). Dataset yang digunakan bernama **iris.csv**, yang berisi data pengukuran morfologi bunga Iris yang terdiri dari empat atribut numerik, yaitu **Sepal Length**, **Sepal Width**, **Petal Length**, dan **Petal Width**, yang masing-masing merepresentasikan panjang dan lebar bagian sepal serta petal bunga dalam satuan sentimeter.

```
df= pd.read_csv('https://drive.google.com/uc?export=download&id=1WBJFWG0T0zRjr_q6KKrjeupx8hID5WAL')
```

3. Melihat Informasi Umum

Langkah ini digunakan untuk mengetahui struktur dan karakteristik awal dari dataset. Fungsi info() menunjukkan tipe data setiap kolom dan jumlah nilai non-null, sedangkan describe() memberikan ringkasan statistik dari kolom numerik.

```
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
```

4. Pemisahan Fitur dan Target

```
X = df.drop('Species', axis=1)
y = df['Species']
```

Dataset dipisahkan menjadi dua bagian, yaitu:

X sebagai fitur (variabel independen) yang berisi atribut sepal dan petal.

y sebagai target (variabel dependen) yang berisi kelas atau spesies bunga Iris.

Pemisahan ini diperlukan agar model dapat mempelajari hubungan antara fitur dan target.

5. Pembagian Data Training dan Testing

   Data dibagi menjadi dua bagian, yaitu: 80% data latih (training) untuk melatih model 20% data uji (testing) untuk menguji performa model

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

   Parameter stratify=y digunakan agar proporsi kelas pada data latih dan data uji tetap seimbang. random_state=42 digunakan untuk memastikan hasil pembagian data dapat direproduksi.

6. Modeling Decision Tree

   criterion='gini' digunakan untuk mengukur tingkat ketidakmurnian data dalam proses pemilihan node. random_state=42 bertujuan untuk menjaga konsistensi hasil pelatihan model.

```
model = DecisionTreeClassifier(
    criterion='gini',
    random_state=42
)

model.fit(X_train, y_train)
```

```
DecisionTreeClassifier
DecisionTreeClassifier(random_state=42)
```

7. Pelatihan Model

   Fungsi fit() digunakan untuk melatih model Decision Tree menggunakan data latih. Pada tahap ini, model mempelajari pola dan hubungan antara fitur sepal dan petal dengan spesies Iris.

8. Prediksi Data testing

   Kode ini digunakan untuk melakukan prediksi kelas spesies Iris berdasarkan data uji. Hasil prediksi kemudian dibandingkan dengan data aktual untuk mengukur performa model.

9. Evaluasi Model

```
y_pred = model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 1.0

Confusion Matrix:
 [[10  0  0]
 [ 0 10  0]
 [ 0  0 10]]

Classification Report:
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        10
Iris-versicolor       1.00      1.00      1.00        10
 Iris-virginica       1.00      1.00      1.00        10

       accuracy                           1.00        30
      macro avg       1.00      1.00      1.00        30
   weighted avg       1.00      1.00      1.00        30
```

a. Akurasi
   Akurasi digunakan untuk mengukur persentase prediksi yang benar dibandingkan dengan keseluruhan data uji

   Nilai akurasi sebesar **1.0 (100%)** menunjukkan bahwa model **Decision Tree mampu mengklasifikasikan seluruh data uji dengan benar**. Artinya, tidak terdapat kesalahan prediksi pada dataset testing yang digunakan.

b. Confusion matrix

   Iris-setosa: 10 data diuji dan seluruhnya diklasifikasikan dengan benar.
   Iris-versicolor: 10 data diuji dan seluruhnya diklasifikasikan dengan benar.
   Iris-virginica: 10 data diuji dan seluruhnya diklasifikasikan dengan benar.
   Tidak terdapat nilai di luar diagonal utama, yang berarti tidak ada kesalahan klasifikasi antar kelas.

c. Classification report
   Precision = 1.00 → Semua prediksi model untuk setiap kelas adalah benar.
   Recall = 1.00 → Model berhasil mendeteksi seluruh data aktual dari setiap kelas.
   F1-score = 1.00 → Keseimbangan sempurna antara precision dan recall.
   Support menunjukkan jumlah data uji pada masing-masing kelas (10 data per kelas).

10. Menampilkan Hasil pengujian

```
hasil_uji = pd.DataFrame({
    'Actual': y_test.values,
    'Predicted': y_pred
})

hasil_uji.head()
```

	Actual	Predicted
0	Iris-setosa	Iris-setosa
1	Iris-virginica	Iris-virginica
2	Iris-versicolor	Iris-versicolor
3	Iris-versicolor	Iris-versicolor
4	Iris-setosa	Iris-setosa

Kode ini digunakan untuk menampilkan perbandingan antara nilai aktual dan hasil prediksi model dalam bentuk tabel, sehingga hasil pengujian dapat dianalisis secara visual.

Link Github : https://github.com/aisyahhana/Machine-Learning