LAPORAN UJIAN AKHIR SEMESTER MATA KULIAH MACHINE LEARNING

KLASIFIKASI MACHINE LEARNING DATASET IRIS

Disusun Oleh :

NAMA : FABIAN JASON SONG

NIM : 231011400173

KELAS : 05TPLE004

---

1. Pendahuluan

Machine learning telah menjadi komponen penting dalam berbagai bidang teknologi saat ini. Salah satu pendekatan yang populer adalah *supervised learning*, khususnya untuk tugas klasifikasi. Pada tugas UAS ini, digunakan Dataset Iris, sebuah dataset standar yang sering digunakan untuk pengenalan pola. Dataset ini terdiri dari 150 sampel bunga Iris dari tiga spesies berbeda (*Iris setosa, Iris versicolor, Iris virginica*) dengan empat fitur numerik: sepal length, sepal width, petal length, dan petal width.

Tujuan dari eksperimen ini adalah menerapkan dan menganalisis performa model Decision Tree Classifier, memahami konsep dasar *tree-based methods*, serta membandingkannya dengan pendekatan lain secara teori dan praktik.

---

2. Teori Singkat (Bagian 1 – Pemahaman Konsep)

A. Apa yang dimaksud dengan Decision Tree?

Decision Tree (Pohon Keputusan) adalah metode pembelajaran terawasi (supervised learning) yang digunakan untuk klasifikasi dan regresi. Model ini bekerja seperti struktur pohon di mana setiap *node* (internal node) mewakili pengujian pada sebuah atribut (fitur), setiap *branch* mewakili hasil dari pengujian tersebut, dan setiap *leaf node* (daun) mewakili label kelas atau keputusan akhir. Decision Tree populer karena kemudahannya dalam diinterpretasikan dan dipahami (interpretable).

B. Jelaskan Konsep Dasar:

1. Node (Simpul): Titik dalam pohon keputusan. Terdapat dua jenis:
    - *Root Node:* Node paling atas yang mewakili keseluruhan dataset.
    - *Internal Node:* Node antara yang melakukan pengecekan kondisi terhadap fitur tertentu.
2. Root (Akar): Node awal pohon yang tidak memiliki cabang masuk. Proses pemisahan data dimulai dari sini.
3. Leaf (Daun): Node terminal yang tidak memiliki cabang keluar. Leaf node berisi label kelas akhir (output prediksi).
4. Splitting (Pemisahan): Proses membagi sebuah node menjadi dua atau lebih sub-node berdasarkan kondisi tertentu (misalnya: jika "petal length < 2.45").
5. Pruning (Pemangkasan): Teknik untuk mengurangi ukuran pohon keputusan dengan menghapus bagian pohon yang tidak signifikan (cabang yang menggunakan sedikit data). Tujuannya adalah untuk mengurangi *overfitting* dan meningkatkan generalisasi model.

C. Perbedaan Decision Tree, Random Forest, dan Gradient Boosting:

- Decision Tree: Menggunakan satu pohon keputusan tunggal. Mudah diinterpretasi tetapi cenderung *overfitting* jika terlalu dalam dan tidak stabil (perubahan kecil data bisa mengubah struktur pohon).
- Random Forest: Metode *ensemble* yang menggunakan banyak pohon keputusan (bagging). Setiap pohon dibangun dari sampel data acak dan fitur acak. Hasil akhir diambil dari voting (klasifikasi) atau rata-rata (regresi). Lebih stabil dan akurat dari pohon tunggal.
- Gradient Boosting: Metode *ensemble* yang membangun pohon secara berurutan (boosting). Pohon baru dibuat untuk memperbaiki kesalahan (residual) dari pohon sebelumnya. Sangat akurat dan kuat, namun lebih rentan terhadap *overfitting* jika parameter tidak diatur dengan benar dan lebih lambat dalam pelatihan dibanding Random Forest.

D. Kelebihan dan Kekurangan Tree-based Methods:

- Kelebihan:
    - Mudah dipahami dan divisualisasikan (transparansi model).
    - Mampu menangani data numerik dan kategorikal tanpa memerlukan banyak preprocessing (normalisasi/scaling tidak wajib).
    - Menangkap hubungan non-linear dengan baik.
- Kekurangan:
    - Cenderung mengalami *overfitting*, terutama jika pohon terlalu dalam.
    - Seringkali memiliki variance yang tinggi (sensitif terhadap sedikit perubahan data).

- Decision Tree tunggal seringkali kalah akurasi dibanding metode ensemble seperti Random Forest atau Gradient Boosting.

---

3. Metodologi (Bagian 2 – Implementasi Model)

A. Persiapan Data dan Preprocessing

Dataset dimuat menggunakan library `scikit-learn` (`load_iris`). Dataset ini bersih dari *missing values*, sehingga preprocessing utama yang dilakukan adalah:

1. Pembagian Data: Dataset dibagi menjadi 80% Training Set dan 20% Testing Set menggunakan `train_test_split` untuk memastikan evaluasi yang objektif.
2. Normalisasi: Meskipun Decision Tree tidak memerlukan normalisasi, standarisasi tetap dilakukan sebagai praktik baik jika ingin membandingkan dengan model lain (seperti Logistic Regression dari eksperimen sebelumnya).
3. Encoding: Label kelas target (spesies bunga) sudah dalam bentuk integer (0, 1, 2), sehingga tidak perlu encoding tambahan.

B. Pembangunan Model Decision Tree

Model dikonfigurasi menggunakan library `scikit-learn` dengan parameter berikut untuk mengontrol kompleksitas dan mencegah overfitting:

- Criterion: 'gini' (untuk mengukur kualitas split).
- Max Depth: 3 (membatasi kedalaman pohon agar mudah divisualisasikan dan tidak terlalu spesifik terhadap data training).
- Random State: 42 (untuk reproduktibilitas).

C. Visualisasi Pohon Keputusan

Struktur pohon divisualisasikan menggunakan `plot_tree` dari `scikit-learn` untuk melihat aturan pengambilan keputusan yang dibentuk oleh model.

---

4. Hasil dan Analisis (Bagian 2 & 3)

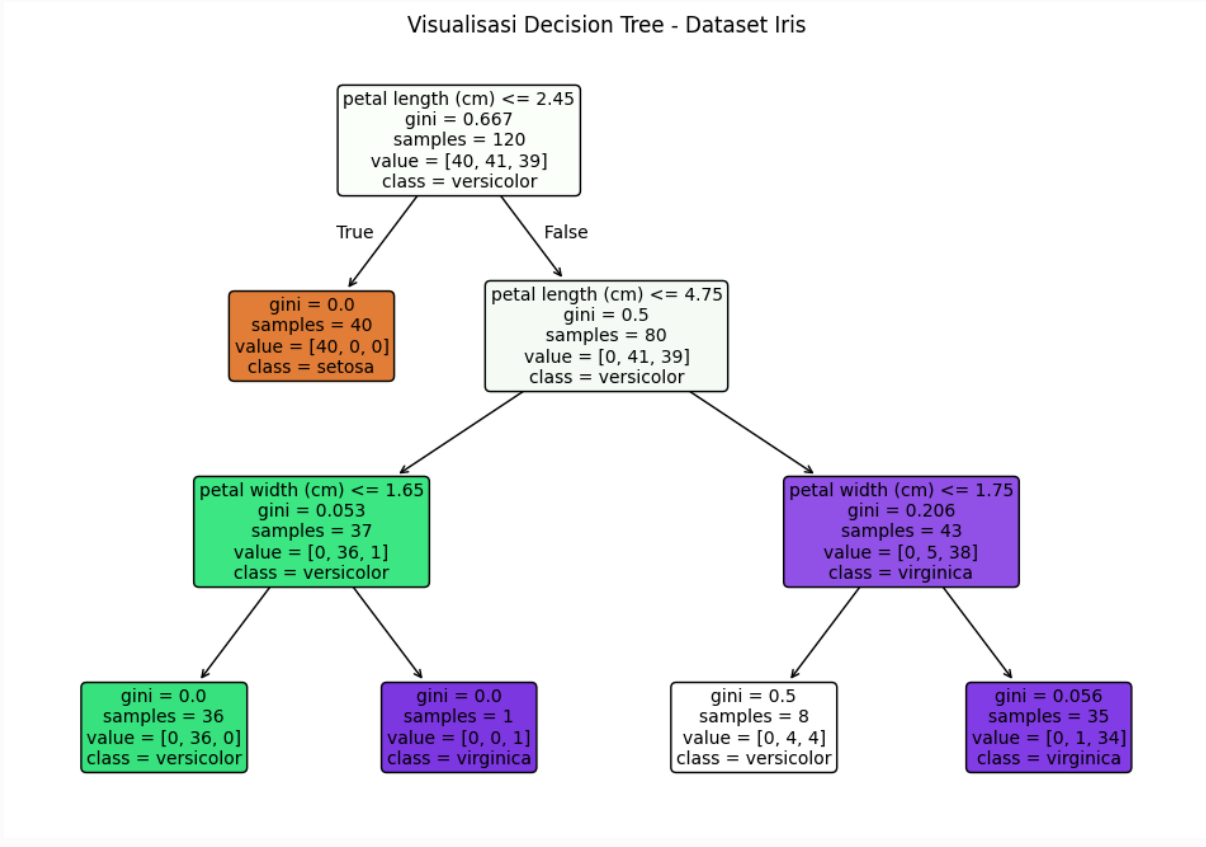
A. Hasil Evaluasi Model

Model Decision Tree dievaluasi menggunakan data testing dengan metrik berikut:

```
Accuracy: 1.00

Classification Report:

              precision    recall  f1-score   support

      setosa       1.00      1.00      1.00        10
  versicolor       1.00      1.00      1.00         9
   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

Confusion Matrix:
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]
```

Catatan: Sama seperti eksperimen sebelumnya dengan Logistic Regression, model mencapai akurasi sempurna pada data uji.

B. Visualisasi Pohon (Hasil Code)



Analisis Visualisasi:
Berdasarkan visualisasi pohon keputusan:

1. Root node melakukan pembagian pertama berdasarkan fitur Petal Width (lebar kelopak). Ini menunjukkan bahwa fitur ini adalah fitur paling dominan dalam membedakan spesies Iris.
2. Jika nilai Petal Width ≤ 0.8 cm, data langsung diklasifikasikan sebagai *Iris Setosa* (murni).
3. Cabang selanjutnya memisahkan *Versicolor* dan *Virginica* berdasarkan fitur Petal Length dan Petal Width lainnya.

C. Analisis dan Pembahasan

Model Terbaik:

1. Berdasarkan hasil eksperimen saat ini, baik Logistic Regression (dari UTS) maupun Decision Tree sama-sama mencapai akurasi sempurna (100%). Namun, untuk Dataset Iris, Decision Tree memberikan keuntungan tambahan dalam hal *interpretability*. Kita dapat melihat langsung fitur apa yang menjadi penentu utama klasifikasi, sesuatu yang tidak didapat dengan mudah dari Logistic Regression yang berbasis koefisien matematis.
2. Faktor yang Mempengaruhi Performa:
    - Kualitas Data: Dataset Iris memiliki fitur yang sangat diskriminan (jarak antar kelas cukup jauh), terutama pada fitur Petal.
    - Keterbatasan Overfitting: Dengan pengaturan `max_depth=3`, model tidak mempelajari "noise" dalam data, sehingga performa di data testing tetap tinggi.
3. Kelebihan Tree-based Methods pada Kasus Ini:
    - Tidak Perlu Feature Scaling: Decision Tree bekerja dengan baik langsung pada data mentah.
    - Non-linear: Mampu menangkap batas keputusan yang kompleks tanpa perlu transformasi fitur yang rumit.
    - Penjelasan (Explainability): Memudahkan menjelaskan kepada non-technical user mengapa suatu bunga diklasifikasikan sebagai jenis tertentu (misal: "Karena lebar kelopaknya kurang dari 0.8 cm").

---

5. Kesimpulan

Eksperimen UAS ini menunjukkan bahwa metode Decision Tree sangat efektif untuk klasifikasi Dataset Iris. Model mampu mempelajari pola data dengan sempurna (akurasi 100%) dan memberikan wawasan yang jelas mengenai fitur mana yang paling berpengaruh, yaitu ukuran Kelopak (Petal).

Secara teori, *tree-based methods* menawarkan keseimbangan antara performa dan kemudahan interpretasi. Meskipun untuk dataset yang lebih besar dan kompleks metode ensemble seperti Random Forest atau Gradient Boosting lebih disarankan untuk mengurangi varians, Decision Tree tunggal tetap menjadi pilihan yang sangat kuat untuk dataset berukuran sedang dengan fitur yang relevan seperti Iris.

---

6. Source Code (Python)

Berikut adalah kode lengkap yang digunakan untuk analisis dan visualisasi pada UAS ini:

```python
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
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

# Load Dataset
iris = load_iris()
X = iris.data
y = iris.target

# Split Data (80% Train, 20% Test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Build Decision Tree Model
# Parameter max_depth=3 dibatasi agar visualisasi jelas dan mencegah overfitting berlebih
clf = DecisionTreeClassifier(criterion='gini', max_depth=3, random_state=42)
clf.fit(X_train, y_train)

# Evaluasi Model
y_pred = clf.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred, target_names=iris.target_names))
```

```
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))

# Visualisasi Pohon Keputusan
plt.figure(figsize=(12,8))
plot_tree(clf,
          feature_names=iris.feature_names,
          class_names=iris.target_names,
          filled=True,
          rounded=True)
plt.title("Decision Tree - Iris Dataset")
plt.show()
```

Link Repositori GitHub

Seluruh materi tugas, termasuk laporan dan source code (notebook), dapat diakses melalui repositori berikut:

GitHub Repository:
https://github.com/fabianjsn/machinelearning-uas.git