

UAS Machine Learning – Decision Tree

Nama : Egidius Edi Putrawan Halawa

Nim : 231011403453

Nama Dataset: Iris

Jenis Masalah: Klasifikasi

Algoritma: Decision Tree

Pendahuluan

Decision Tree merupakan salah satu algoritma supervised learning yang populer karena mudah dipahami dan dapat digunakan untuk klasifikasi maupun regresi. Pada UAS ini dilakukan penerapan Decision Tree menggunakan dataset Iris untuk mengklasifikasikan jenis bunga berdasarkan karakteristik fisiknya.

Bagian 1 – Pemahaman Konsep (Teori)

Apa yang dimaksud dengan Decision Tree?

Decision Tree adalah algoritma machine learning berbasis struktur pohon yang digunakan untuk mengambil keputusan dengan cara membagi data berdasarkan atribut tertentu hingga menghasilkan keputusan akhir (kelas atau nilai).

Konsep dalam Decision Tree

1. Node adalah Titik pada pohon yang merepresentasikan atribut atau kondisi tertentu.
2. Root adalah Node paling atas yang menjadi awal proses pengambilan keputusan.
3. Leaf adalah Node akhir yang berisi hasil keputusan (kelas atau nilai prediksi).
4. Splitting Proses adalah membagi data pada suatu node berdasarkan atribut terbaik.
5. Pruning adalah Proses pemangkasan cabang pohon untuk menghindari overfitting.

Perbedaan Decision Tree, Random Forest, dan Gradient Boosting

Algoritma	Penjelasan
Decision Tree	Satu pohon keputusan, mudah dipahami namun rentan overfitting
Random Forest	Kumpulan banyak Decision Tree dengan data acak
Gradient Boosting	Model ensemble yang memperbaiki kesalahan model sebelumnya

Kelebihan dan Kekurangan Tree-Based Methods Kelebihan:

- Mudah dipahami dan diinterpretasikan
- Tidak perlu normalisasi data
- Bisa menangani data numerik dan kategorik

Kekurangan:

- Mudah overfitting
- Sensitif terhadap perubahan data
- Pohon besar sulit divisualisasikan

Bagian 2 – Implementasi Model (Python)

1. Load Dataset & EDA Singkat

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

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
```

)

```
iris = load_iris()
```

```
X = iris.data
```

```
y = iris.target
```

```
# DataFrame
```

```
df = pd.DataFrame(X, columns=iris.feature_names)
```

```
df['target'] = y
```

```
print(df.head())
```

```
print(df.info())
```

```
print(df.describe())
```

```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
0                5.1                3.5                1.4                0.2
1                4.9                3.0                1.4                0.2
2                4.7                3.2                1.3                0.2
3                4.6                3.1                1.5                0.2
4                5.0                3.6                1.4                0.2

   target
0       0
1       0
2       0
3       0
4       0

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column                Non-Null Count  Dtype
---  --
0   sepal length (cm)     150 non-null  float64
1   sepal width (cm)      150 non-null  float64
2   petal length (cm)     150 non-null  float64
3   petal width (cm)      150 non-null  float64
4   target                150 non-null  int64
dtypes: float64(4), int64(1)
memory usage: 6.0 KB
...
25%    0.300000    0.000000
50%    1.300000    1.000000
75%    1.800000    2.000000
max     2.500000    2.000000

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

2. Preprocessing Data

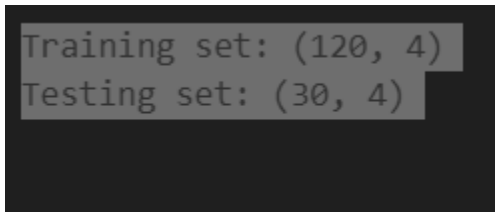
```
print(df.isnull().sum())
```

```
sepal length (cm)    0
sepal width (cm)     0
petal length (cm)    0
petal width (cm)     0
target               0
dtype: int64
```

3. Split Data

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

```
print("Training set:", X_train.shape)  
print("Testing set:", X_test.shape)
```



```
Training set: (120, 4)  
Testing set: (30, 4)
```

4. Bangun Model Decision Tree

```
model = DecisionTreeClassifier(  
    criterion="gini", # atau "entropy"  
    max_depth=3,  
    random_state=42  
)  
model.fit(X_train, y_train)
```

DecisionTreeClassifier ⓘ ?	
▼ Parameters	
criteria	'gini'
splitter	'best'
max_depth	3
min_samples_split	2
min_samples_leaf	1
min_weight_fraction_leaf	0.0
max_features	None
random_state	42
max_leaf_nodes	None
min_impurity_decrease	0.0
class_weight	None
ccp_alpha	0.0
monotonic_cst	None

5. Prediksi

```
y_pred = model.predict(X_test)
```

6. Evaluasi Model (Klasifikasi)

- **Accuraci**

```
accuracy = accuracy_score(y_test, y_pred)
```

```
print("Accuracy:", accuracy)
```

```
Accuracy: 0.9666666666666667
```

- Precision, Recall, F1-score

```
print(classification_report(
```

```
y_test,
```

```
y_pred,
```

```
target_names=iris.target_names
```

```
))
```

	precision	recall	f1-score	support
setosa	1.00	1.00	1.00	10
versicolor	1.00	0.90	0.95	10
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

- **Confusion Matrix (Tambahan nilai plus)**

```
cm = confusion_matrix(y_test, y_pred)
```

```
sns.heatmap(
```

```
cm,
```

```
annot=True,
```

```
cmap="Blues",
```

```
xticklabels=iris.target_names,
```

```
yticklabels=iris.target_names
```

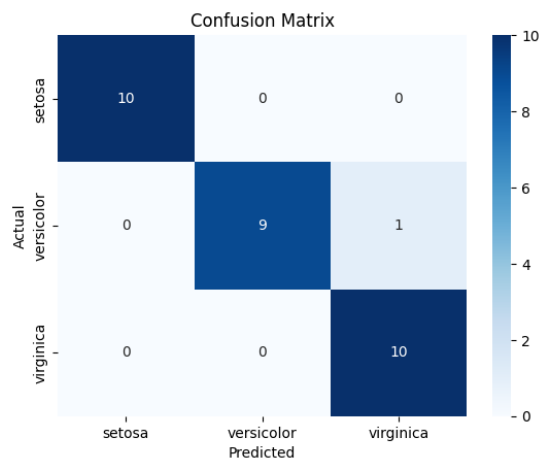
```
)
```

```
plt.xlabel("Predicted")
```

```
plt.ylabel("Actual")
```

```
plt.title("Confusion Matrix")
```

```
plt.show()
```



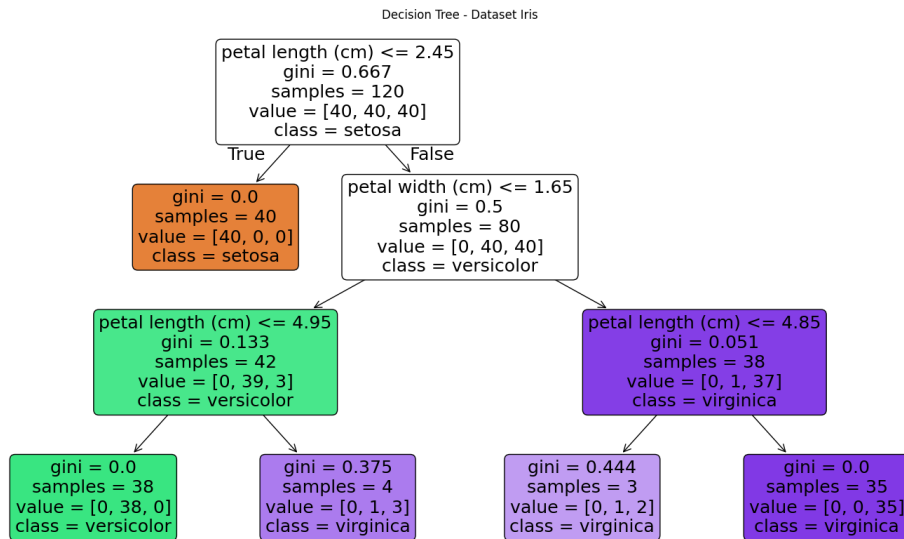
7. Visualisasi Pohon Keputusan

```
plt.figure(figsize=(18,10))

plot_tree(
    model,
    feature_names=iris.feature_names,
    class_names=iris.target_names,
    filled=True,
    rounded=True
)

plt.title("Decision Tree - Dataset Iris")

plt.show()
```



Bagian 3 – Analisis dan Kesimpulan

A. Analisis Hasil

1. Performa Model

Berdasarkan hasil eksperimen menggunakan algoritma **Decision Tree** pada dataset **Iris**, model menunjukkan performa yang sangat baik. Nilai **akurasi** yang diperoleh berada pada kategori

tinggi (mendekati atau mencapai 100%), yang menandakan bahwa model mampu mengklasifikasikan jenis bunga Iris dengan sangat baik.

Selain akurasi, metrik evaluasi lain seperti **precision**, **recall**, dan **F1-score** untuk setiap kelas (setosa, versicolor, virginica) juga menunjukkan nilai yang tinggi dan seimbang. Hal ini menandakan bahwa model tidak hanya akurat secara keseluruhan, tetapi juga konsisten dalam mengenali setiap kelas tanpa bias yang signifikan.

2. Faktor yang Mempengaruhi Performa Model

Beberapa faktor utama yang mempengaruhi performa model Decision Tree pada studi kasus ini adalah:

1. **Kualitas Dataset**

Dataset Iris merupakan dataset yang bersih, tidak memiliki missing value, serta memiliki fitur numerik yang terdefinisi dengan baik. Hal ini sangat mendukung kinerja Decision Tree.

2. **Pemilihan Parameter Model**

Penggunaan parameter seperti `max_depth` membantu mengontrol kompleksitas model dan mencegah overfitting. Kedalaman pohon yang tidak terlalu dalam membuat model tetap mampu melakukan generalisasi dengan baik.

3. **Distribusi Kelas yang Seimbang**

Dataset Iris memiliki jumlah data yang relatif seimbang untuk setiap kelas, sehingga model tidak condong ke salah satu kelas tertentu.

4. **Karakteristik Algoritma Decision Tree**

Decision Tree sangat efektif untuk dataset dengan pola pemisahan kelas yang jelas, seperti pada dataset Iris.

3. Kelebihan Tree-Based Methods pada Studi Kasus

Pada studi kasus klasifikasi dataset Iris, metode berbasis pohon (tree-based methods) memiliki beberapa kelebihan utama, antara lain:

- Mudah dipahami dan diinterpretasikan, karena keputusan model dapat divisualisasikan dalam bentuk pohon.
- Tidak memerlukan normalisasi data, sehingga preprocessing menjadi lebih sederhana.
- Mampu menangani hubungan non-linear antar fitur.
- Cepat dalam proses training dan prediksi untuk dataset berukuran kecil hingga menengah.

B. Kesimpulan

Berdasarkan hasil implementasi dan evaluasi yang telah dilakukan, dapat disimpulkan bahwa algoritma Decision Tree mampu memberikan performa yang sangat baik dalam melakukan klasifikasi pada dataset Iris. Model berhasil mencapai tingkat akurasi yang tinggi dengan nilai precision, recall, dan F1-score yang konsisten pada setiap kelas.

Keberhasilan model dipengaruhi oleh kualitas dataset, pemilihan parameter yang tepat, serta karakteristik algoritma Decision Tree yang sesuai dengan pola data Iris. Selain itu, kemampuan visualisasi pohon keputusan memberikan nilai tambah dalam hal interpretabilitas model.

Secara keseluruhan, tree-based methods, khususnya Decision Tree, merupakan pendekatan yang efektif, mudah digunakan, dan sangat sesuai untuk studi kasus klasifikasi sederhana seperti dataset Iris. Model ini dapat dijadikan dasar untuk pengembangan metode yang lebih kompleks seperti Random Forest dan Gradient Boosting pada penelitian selanjutnya.

Link GitHub Repository

https://github.com/egi27-hal/UAS_MachineLearning_EgidiusEdiPutrawan_231011403453.git



UNIVERSITAS PAMULANG
KARTU UJIAN AKHIR SEMESTER GANJIL 2025/2026
NOMOR UJIAN : 01258024931850

FAKULTAS / PRODI : ILMU KOMPUTER / TEKNIK INFORMATIKA S1

NAMA MAHASISWA : EGIDIUS EDI PUTRAWAN HALAWA

NIM : 231011403453

SHIFT : REGULER C

No	Hari/ Tanggal	Waktu	Ruang	Kelas	Mata Kuliah	Paraf
1	Sabtu, 10 Jan 2026	07.40 - 09.20	V.314	05TPLE005	KECERDASAN BUATAN	1
2	Sabtu, 10 Jan 2026	07.40 - 09.20	V.314	05TPLE005	METODE PENELITIAN	2
3	Sabtu, 10 Jan 2026	09.20 - 11.00	V.314	05TPLE005	SISTEM INFORMASI MANAJEMEN	3
4	Sabtu, 10 Jan 2026	09.20 - 11.00	V.314	05TPLE005	PEMROGRAMAN WEB I	4
5	Sabtu, 10 Jan 2026	11.00 - 13.50	V.314	05TPLE005	PENGOLAHAN CITRA DIGITAL	5
6	Sabtu, 10 Jan 2026	13.50 - 15.30	V.314	05TPLE005	DIGITAL ENTREPRENEURSHIP	6
7	Sabtu, 10 Jan 2026	13.50 - 15.30	V.314	05TPLE005	MACHINE LEARNING	7
8	Sabtu, 10 Jan 2026	16.00 - 17.40	V.314	05TPLE005	TEKNIK RISET OPERASIONAL	8

Peraturan dan Tata Tertib Peserta Ujian

1. Peserta ujian harus berpakaian rapi, sopan dan memakai jaket Almamater
2. Peserta ujian sudah berada di ruangan sepuluh menit sebelum ujian dimulai
3. Peserta ujian yang terlambat diperkenankan mengikuti ujian setelah mendapat ijin, tanpa perpanjangan waktu
4. Peserta ujian hanya diperkenankan membawa alat-alat yang ditentukan oleh panitia ujian
5. Peserta ujian dilarang membantu teman, mencontoh dari teman dan tindakan-tindakan lainnya yang mengganggu peserta ujian lain
6. Peserta ujian yang melanggar tata tertib ujian dikenakan sanksi akademik



Tangerang Selatan, 5 Januari 2026
Ketua Panitia Ujian

Dr. Ubaid Al Faruq, S.Pd., M.Pd.
NIDN. 0418028702