

Nama: Juan Montoya Darmawan

NIM: 231011403196

Kelas: 05TPLE013

Mata Kuliah: Kecerdasan Buatan

JAWABAN UAS Kecerdasan Buatan

## Pendahuluan

Perkembangan kecerdasan buatan (Artificial Intelligence) telah mendorong pemanfaatan machine learning dalam berbagai bidang, salah satunya dalam analisis data dan prediksi. Machine learning memungkinkan komputer untuk mempelajari pola dari data historis dan menghasilkan prediksi secara otomatis. Salah satu algoritma yang sering digunakan dalam permasalahan klasifikasi adalah Decision Tree.

Decision Tree memiliki keunggulan dalam hal interpretabilitas karena struktur model dapat divisualisasikan dalam bentuk pohon keputusan yang mudah dipahami. Pada penelitian ini, algoritma Decision Tree diterapkan untuk memprediksi tingkat kelangsungan hidup penumpang kapal Titanic berdasarkan atribut penumpang seperti kelas penumpang, usia, jenis kelamin, dan jumlah keluarga yang ikut berlayar.

Tujuan dari penelitian ini adalah untuk membangun model klasifikasi menggunakan Decision Tree, mengevaluasi performa model, serta menganalisis faktor-faktor yang memengaruhi hasil prediksi.

## Teori Singkat

Decision Tree adalah algoritma supervised learning yang digunakan untuk klasifikasi maupun regresi, dengan cara memodelkan keputusan dalam bentuk struktur pohon. Model ini membagi data berdasarkan atribut tertentu secara bertahap hingga menghasilkan keputusan akhir pada daun (leaf). Setiap percabangan merepresentasikan aturan keputusan (if–then) yang mudah dipahami oleh manusia.

### 1. Node

Node adalah titik pada pohon yang merepresentasikan fitur atau kondisi tertentu yang digunakan untuk membagi data.

### 2. Root

Root adalah node paling atas yang menjadi titik awal pemisahan data. Biasanya dipilih fitur dengan informasi paling tinggi.

### 3. Leaf

Leaf adalah node akhir yang tidak memiliki cabang lagi dan berisi hasil prediksi (kelas atau nilai).

### 4. Splitting

Splitting adalah proses membagi data ke dalam beberapa subset berdasarkan nilai fitur tertentu agar data menjadi lebih homogen.

### 5. Pruning

Pruning adalah proses memangkas cabang pohon untuk mengurangi overfitting dan meningkatkan generalisasi model.

Decision Tree merupakan algoritma pembelajaran mesin yang menggunakan satu struktur pohon keputusan untuk melakukan prediksi, sehingga mudah dipahami dan diinterpretasikan, namun memiliki kelemahan yaitu rentan terhadap overfitting. Random Forest merupakan pengembangan dari Decision Tree yang menggabungkan banyak pohon keputusan (ensemble) yang dilatih secara acak menggunakan teknik bagging, sehingga mampu meningkatkan stabilitas model dan akurasi prediksi serta mengurangi risiko overfitting, meskipun interpretasinya menjadi lebih kompleks. Sementara itu, Gradient Boosting membangun model secara bertahap, di mana setiap pohon baru berfungsi untuk memperbaiki kesalahan dari model sebelumnya, sehingga mampu menghasilkan performa prediksi yang sangat tinggi, tetapi membutuhkan waktu komputasi lebih besar dan sensitif terhadap pengaturan parameter. Dengan demikian, pemilihan algoritma harus disesuaikan dengan kebutuhan interpretabilitas, kompleksitas data, dan sumber daya komputasi yang tersedia.

## Kelebihan dan Kekurangan Tree-Based Methods

### Kelebihan

- Mudah dipahami dan divisualisasikan.
- Tidak memerlukan normalisasi data.
- Bisa menangani data numerik dan kategorikal.
- Mampu menangani hubungan non-linear.

### Kekurangan

- Rentan terhadap overfitting.
- Sensitif terhadap perubahan data kecil.
- Model ensemble sulit diinterpretasikan.
- Bisa membutuhkan komputasi lebih besar.

## Metodologi

### Dataset

Dataset yang digunakan adalah Titanic Dataset, yang berisi data penumpang kapal Titanic dengan target klasifikasi apakah penumpang selamat atau tidak. Dataset terdiri dari beberapa atribut seperti kelas penumpang, usia, jenis kelamin, tarif tiket, dan jumlah keluarga.

## **Tahapan Penelitian**

Tahapan penelitian meliputi:

1. Load Dataset

Dataset dimuat menggunakan library pandas.

2. Exploratory Data Analysis (EDA)

Dilakukan pengecekan struktur data, missing value, dan statistik deskriptif.

3. Preprocessing Data

- Menghapus kolom teks yang tidak relevan (name).
- Melakukan encoding pada fitur kategorikal (sex).
- Memastikan seluruh data dalam bentuk numerik.

4. Split Data

Data dibagi menjadi training set (80%) dan testing set (20%).

5. Pembangunan Model

Model Decision Tree dibangun menggunakan parameter:

- criterion = gini
- max\_depth = 4

6. Evaluasi Model

Evaluasi dilakukan menggunakan metrik:

- Accuracy
- Precision
- Recall
- F1-score

7. Visualisasi Model

Struktur pohon divisualisasikan menggunakan fungsi plot\_tree.

## **Hasil & Analisis**

Berdasarkan hasil eksperimen, model Decision Tree mampu menghasilkan nilai akurasi sebesar 0.7552447552447552. Nilai precision, recall, dan F1-score menunjukkan bahwa model cukup baik dalam membedakan penumpang yang selamat dan tidak selamat.

Faktor yang memengaruhi performa model antara lain:

- Kualitas dataset dan jumlah fitur yang digunakan.
- Pemilihan parameter max\_depth yang mengontrol kompleksitas model.
- Distribusi kelas target yang tidak seimbang.
- Jumlah data training.

Visualisasi pohon keputusan menunjukkan bahwa fitur seperti jenis kelamin, kelas penumpang, dan usia memiliki pengaruh besar dalam proses pengambilan keputusan model.

## Kesimpulan

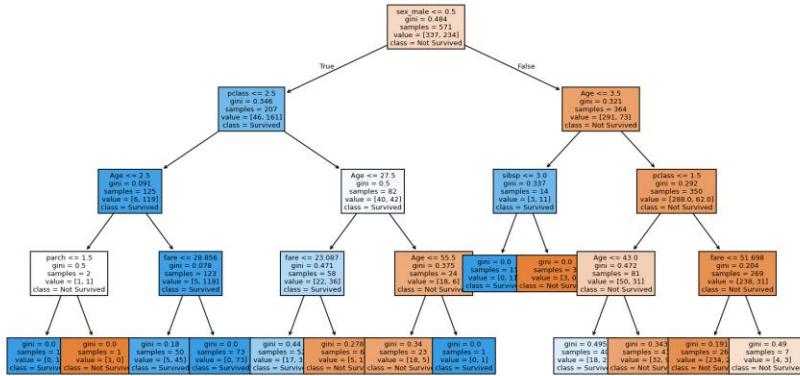
Penerapan algoritma Decision Tree pada dataset Titanic berhasil dilakukan dengan performa klasifikasi yang cukup baik. Model mudah diinterpretasikan dan dapat divisualisasikan dengan jelas. Decision Tree cocok digunakan sebagai model dasar dalam permasalahan klasifikasi, namun memiliki kelemahan terhadap overfitting sehingga diperlukan pengaturan parameter atau penggunaan metode ensemble untuk meningkatkan performa di masa mendatang.

# Source Code & Gambar Output

```

1 # 1. Import Library
2 import pandas as pd
3 import numpy as np
4 import matplotlib.pyplot as plt
5
6 from sklearn.model_selection import train_test_split
7 from sklearn.tree import DecisionTreeClassifier, plot_tree
8 from sklearn.metrics import accuracy_score, classification_report
9
10
11 # 2. Load Dataset
12 df = pd.read_csv("titanic.csv")
13
14 print(df.head())
15 print(df.info())
16
17 # 3. EDA Singkat
18 print(df.isnull().sum())
19 print(df.describe())
20
21
22 # 4. Preprocessing
23 # Handle missing value
24 df['Age'].fillna(df['Age'].median(), inplace=True)
25 df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)
26
27 # Encoding categorical
28 df = pd.get_dummies(df, columns=['Sex', 'Embarked'], drop_first=True)
29
30 # Drop kolom yang tidak relevan
31 df.drop(['Name', 'Ticket', 'Cabin'], axis=1, inplace=True)
32
33 # 5. Split Data
34 X = df.drop('Survived', axis=1)
35 y = df['Survived']
36
37 X_train, X_test, y_train, y_test = train_test_split(
38     X, y, test_size=0.2, random_state=42
39 )
40
41 # 6. Build Decision Tree Model
42 model = DecisionTreeClassifier(
43     criterion='gini',
44     max_depth=4,
45     random_state=42
46 )
47
48 model.fit(X_train, y_train)
49
50 # 7. Evaluate
51 y_pred = model.predict(X_test)
52
53 print("Accuracy:", accuracy_score(y_test, y_pred))
54 print(classification_report(y_test, y_pred))
55
56 # 8. Visualisasi
57 plt.figure(figsize=(20,10))
58 plot_tree(model, feature_names=X.columns, class_names=["Not Survived", "Survived"], filled=True)
59 plt.show()

```



```

PS D:\project\kecerdasan buatan> & "D:/project/kecerdasan buatan/venv/Scripts/Activate.ps1"
(venv) PS D:\project\kecerdasan buatan> & 'd:\project\kecerdasan buatan\venv\Scripts\python.exe' 'c:\Users\user\venv\Scripts\debugpy\launcher' '49885' '--' 'D:\project\kecerdasan buatan\UAS\uas.py'
   survived  pclass          name    sex   Age   fare  sibsp  parch
0         0      3  Braund, Mr. Owen Harris   male  22.0  7.2500     1     0
1         1      1  Cumings, Mrs. John Bradley  female  38.0  71.2833     1     0
2         1      3        Heikkinen, Miss. Laina  female  26.0  7.9250     0     0
3         1      1  Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0  53.1000     1     0
4         0      3       Allen, Mr. William Henry   male  35.0  8.0500     0     0
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 714 entries, 0 to 713
Data columns (total 8 columns):
 #   Column   Non-Null Count  Dtype  
--- 
 0   survived   714 non-null    int64  
 1   pclass     714 non-null    int64  
 2   name       714 non-null    object  
 3   sex        714 non-null    object  
 4   Age        714 non-null    float64 
 5   fare       714 non-null    float64 
 6   sibsp      714 non-null    int64  
 7   parch      714 non-null    int64  
dtypes: float64(2), int64(4), object(2)
memory usage: 44.8+ KB
None

Jumlah missing value:
survived    0
pclass      0
name        0
sex         0
Age         0
fare         0
sibsp       0
parch       0
dtype: int64

Statistik deskriptif:

```

	survived	pclass	Age	fare	sibsp	parch
count	714.000000	714.000000	714.000000	714.000000	714.000000	714.000000
mean	0.406162	2.236695	29.699118	34.694514	0.512605	0.431373
std	0.491460	0.838250	14.526497	52.918930	0.929783	0.853289
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	1.000000	20.125000	8.050000	0.000000	0.000000
50%	0.000000	2.000000	28.000000	15.741700	0.000000	0.000000
75%	1.000000	3.000000	38.000000	33.375000	1.000000	1.000000
max	1.000000	3.000000	80.000000	512.329200	5.000000	6.000000

Dataset setelah preprocessing:

	survived	pclass	Age	fare	sibsp	parch	sex_male
0	0	3	22.0	7.2500	1	0	True
1	1	1	38.0	71.2833	1	0	False
2	1	3	26.0	7.9250	0	0	False
3	1	1	35.0	53.1000	1	0	False
4	0	3	35.0	8.0500	0	0	True

Accuracy: 0.7552447552447552

Classification Report:

	precision	recall	f1-score	support
0	0.80	0.79	0.80	87
1	0.68	0.70	0.69	56
accuracy			0.76	143
macro avg	0.74	0.74	0.74	143
weighted avg	0.76	0.76	0.76	143

Link:

[https://github.com/cucumbees/uas\\_kecerdasanbuatan](https://github.com/cucumbees/uas_kecerdasanbuatan)