

Nama : Maria Nagita Tria Vanessa
 Nim : 231011400228
 Kelas : TPLE005
 Mata Kuliah : Machine Learning (UAS)

Pemahaman Konsep (Teori)

1. Apa yang dimaksud dengan Decision Tree?

Jawab: Decision Tree adalah algoritma pembelajaran mesin yang membentuk struktur pohon untuk mengambil keputusan berdasarkan aturan-aturan tertentu. Setiap keputusan dibuat dengan membagi data berdasarkan fitur yang paling informatif hingga mencapai keputusan akhir pada daun (leaf). Bayangkan Decision Tree seperti cara manusia berpikir saat memilih sesuatu. Kita biasanya bertanya dari hal yang paling umum, lalu semakin spesifik, sampai mendapatkan jawaban. Misalnya: "Apakah cuaca hujan?", jika ya maka "bawa payung", jika tidak maka "tidak perlu payung". Pola bertanya inilah yang ditiru oleh Decision Tree. Cara kerja Decision Tree dapat dijelaskan sebagai sebuah proses bertahap yang menyerupai cara manusia berpikir ketika mengambil keputusan. Pada awalnya, sistem akan melihat seluruh data yang tersedia dan mencari satu atribut yang paling berpengaruh dalam membedakan data. Atribut inilah yang dipilih sebagai akar (root) dari pohon keputusan, karena mampu membagi data ke dalam kelompok yang paling jelas perbedaannya. Setelah atribut tersebut dipilih, data kemudian dipisahkan ke dalam beberapa cabang sesuai dengan nilai atau kondisi dari atribut tersebut. Setiap cabang mewakili hasil dari suatu keputusan, misalnya "ya" atau "tidak", atau rentang nilai tertentu. Pada tahap ini, data yang awalnya menyatu mulai terbagi menjadi kelompok-kelompok yang lebih kecil dan lebih spesifik. Proses pemilahan ini tidak berhenti di satu tahap saja. Pada setiap cabang yang terbentuk, sistem kembali melakukan proses yang sama, yaitu memilih atribut terbaik berikutnya untuk memisahkan data pada cabang tersebut. Proses ini terus berulang hingga data dalam satu cabang sudah memiliki hasil keputusan yang sama, sehingga tidak perlu dibagi lagi, atau ketika tidak ada atribut tersisa yang dapat digunakan untuk melakukan pemisahan. Pada akhirnya, seluruh proses ini menghasilkan sekumpulan aturan keputusan yang jelas dan mudah dipahami. Aturan tersebut biasanya berbentuk pernyataan logis IF-THEN, misalnya jika suatu kondisi terpenuhi maka keputusan tertentu akan diambil. Dengan cara kerja seperti ini, Decision Tree mampu memberikan hasil prediksi sekaligus penjelasan yang logis dan terstruktur.

2. Jelaskan Konsep

- Node: Node adalah titik dalam pohon yang merepresentasikan fitur atau atribut yang digunakan untuk melakukan keputusan.
- Root: Root adalah node paling atas dalam pohon yang menjadi titik awal pengambilan keputusan.
- Leaf: Leaf adalah node akhir yang berisi hasil prediksi atau kelas akhir.
- Splitting: Splitting adalah proses membagi data ke dalam cabang-cabang berdasarkan nilai fitur tertentu untuk meningkatkan homogenitas data.
- Pruning: Pruning adalah proses pemangkasan cabang pohon yang tidak signifikan untuk mencegah overfitting dan meningkatkan kemampuan generalisasi model.

3. Perbedaan Decision Tree, Random Forest, dan Gradient Boosting

| Algoritma | Penjelasan |
|-------------------|--|
| Decision Tree | Satu pohon keputusan tunggal |
| Random Forest | Kumpulan banyak decision tree dengan teknik bagging |
| Gradient Boosting | Pohon dibangun bertahap untuk memperbaiki kesalahan sebelumnya |

4. Kelebihan dan Kekurangan Tree-Based Methods

Jawab:

Kelebihan:

- Mudah dipahami dan diinterpretasikan
- Tidak memerlukan normalisasi data
- Dapat memahami data numerik dan kategorikal

Kekurangan

- Rentan terhadap overfitting
- Sensitif terhadap perubahan data
- Performa tunggal biasanya kalah dibanding ensemble methods

Implementasi Model (Python)

1. Load dan eksplorasi Dataset (EDA)

Dataset : Titanic dataset (kaggle)

Target : *Survived* (0=Tidak selamat, 1=Selamat)

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

data = pd.read_csv("titanic.csv")
data.head()
```

Informasi Dataset:

```
data.info()
data.isnull().sum()
```

2. Preprocessing Data

```
from sklearn.preprocessing import LabelEncoder

# Mengisi missing value
data['Age'].fillna(data['Age'].median(), inplace=True)
data['Embarked'].fillna(data['Embarked'].mode()[0], inplace=True)

# Encoding data kategorikal
encoder = LabelEncoder()
data['Sex'] = encoder.fit_transform(data['Sex'])
data['Embarked'] = encoder.fit_transform(data['Embarked'])

# Seleksi fitur
X = data[['Pclass', 'Sex', 'Age', 'Fare', 'Embarked']]
y = data['Survived']
```

3. Split data training dan testing

```
from sklearn.model_selection import train_test_split

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

4. Membangun model decision tree

```
from sklearn.tree import DecisionTreeClassifier

model = DecisionTreeClassifier(
    criterion='gini',
    max_depth=5,
    random_state=42
)

model.fit(X_train, y_train)
```

5. Evaluasi Model

```
from sklearn.metrics import accuracy_score, classification_report

y_pred = model.predict(X_test)
```

```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
print(classification_report(y_test, y_pred))
```

Analisis dan Kesimpulan

1. Model Terbaik

Model decision tree dengan parameter:

- Criterion: gini
- max_depth: 5

Menunjukkan performa terbaik dengan kesinambungan antara akurasi dan generalisasi

2. Faktor yang mempengaruhi performa model

- Pemilihan fitur yang relevan
- Penanganan missing value
- Kedalaman pohon (max_depth)
- Distribusi data training dan testing

3. Kelebihan Tree-Based Methods pada studi kasus titanic

- Mudah menjelaskan faktor keselamatan penumpang
- Visualisasi membantu analisis keputusan
- Cocok untuk dataset campuran numerik dan kategorikal

4. Kesimpulan Akhir

Decision Tree terbukti efektif dalam memprediksi kelangsungan hidup penumpang Titanic. Model ini memberikan hasil yang cukup baik dengan interpretasi yang jelas. Meskipun rentan terhadap overfitting, penggunaan pruning dan pembatasan kedalaman pohon dapat meningkatkan performa model secara signifikan.