

LAPORAN ULANGAN AKHIR SEMESTER (UAS)
MATA KULIAH MACHINE LEARNING
PENERAPAN ALGORITMA DECISION TREE DALAM
KLASIFIKASI PERFORMA SISWA

Dosen Pengampu:
AGUNG PERDANANTO S.Kom, M.Kom



Oleh:
Saviola Dwi Saputra
231011400200
05TPLE004

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PAMULANG
TANGERANG SELATAN
2026

BAB I

PENDAHULUAN

1.1. Latar Belakang

Machine Learning banyak digunakan untuk membantu pengambilan keputusan berbasis data, termasuk di bidang pendidikan. Salah satu algoritma yang sering digunakan untuk klasifikasi adalah Decision Tree karena mudah dipahami dan diinterpretasikan. Dataset Student Performance digunakan untuk memprediksi performa siswa berdasarkan data akademik dan non-akademik.

1.2. Rumusan Masalah

Langkah-langkah yang dilakukan dalam eksperimen ini adalah:

1. Bagaimana cara menerapkan algoritma Decision Tree pada Dataset Student Performance?.
2. Seberapa baik performa model dalam memprediksi performa siswa?.

1.3. Tujuan Penelitian

Menerapkan algoritma Decision Tree untuk memprediksi performa siswa dan mengevaluasi hasil model klasifikasi.

1.4. Batasan Masalah

- Dataset yang digunakan adalah Student Performance
- Algoritma yang digunakan hanya Decision Tree
- Evaluasi menggunakan metrik klasifikasi

1.5. Manfaat Penelitian

Memberikan gambaran penerapan Decision Tree serta membantu memahami faktor yang memengaruhi performa siswa.

BAB II

DASAR TEORI

2.1. Machine Learning

Machine Learning adalah cabang kecerdasan buatan yang memungkinkan sistem belajar dari data tanpa pemrograman eksplisit.

2.2. Decision Tree

Decision Tree adalah algoritma klasifikasi yang menggunakan struktur pohon untuk pengambilan keputusan.

2.3. Konsep Decision Tree

- **2.3.1 Node:** Titik pengujian atribut
- **2.3.2 Root:** Node awal
- **2.3.3 Leaf:** Hasil prediksi
- **2.3.4 Splitting:** Proses pembagian data
- **2.3.5 Pruning:** Pemangkasan pohon untuk mencegah overfittin

2.4. Perbandingan Metode

- **Decision Tree:** Mudah dipahami, rawan overfitting
- **Random Forest:** Gabungan banyak pohon, lebih stabil
- **Gradient Boosting:** Pohon dibangun bertahap untuk meningkatkan akurasi

2.5. Kelebihan dan Kekurangan

Kelebihan: Interpretatif, fleksibel

Kekurangan: Overfitting, kurang stabi

BAB III

METODOLOGI PENELITIAN

3.1 Dataset Student Performance

Dataset Student Performance merupakan dataset yang berisi informasi akademik dan non-akademik siswa, seperti jenis kelamin, usia, latar belakang keluarga, kebiasaan belajar, serta nilai akhir siswa. Dataset ini digunakan untuk memprediksi performa siswa berdasarkan faktor-faktor tersebut. Pada penelitian ini, performa siswa diklasifikasikan menjadi dua kelas, yaitu lulus dan tidak lulus.

3.2 Tahapan Penelitian

Tahapan penelitian dilakukan secara sistematis, dimulai dari pengumpulan dataset, eksplorasi data awal, preprocessing data, pembangunan model Decision Tree, hingga evaluasi performa model. Tahapan ini bertujuan untuk memastikan data yang digunakan layak dan model yang dihasilkan memiliki performa yang baik.

3.3 Preprocessing Data

Preprocessing data dilakukan untuk meningkatkan kualitas data sebelum digunakan dalam proses pelatihan model. Tahapan preprocessing meliputi:

- Menghilangkan atau mengatasi nilai yang tidak relevan
- Mengubah data kategorikal menjadi data numerik menggunakan teknik encoding
- Membuat label target klasifikasi berdasarkan nilai akhir siswa

Tahap ini sangat penting karena kualitas data sangat memengaruhi hasil model.

3.4 Pembagian Data (Training & Testing)

Dataset dibagi menjadi dua bagian, yaitu data latih dan data uji. Data latih digunakan untuk membangun model Decision Tree, sedangkan data uji digunakan untuk mengukur kemampuan model dalam melakukan prediksi pada data yang belum pernah dilihat sebelumnya. Pembagian data dilakukan untuk menghindari bias dan overfitting.

3.5 Pembangunan Model Decision Tree

Model dibangun menggunakan algoritma Decision Tree dari library scikit-learn. Parameter penting seperti `max_depth` digunakan untuk membatasi kedalaman pohon agar model tidak terlalu kompleks. Pemilihan parameter ini bertujuan untuk menghasilkan model yang seimbang antara akurasi dan kemampuan generalisasi.

3.6 Evaluasi Model

Evaluasi model dilakukan untuk mengukur performa Decision Tree dalam mengklasifikasikan performa siswa. Metrik evaluasi yang digunakan meliputi accuracy, precision, recall, dan F1-score. Metrik ini memberikan gambaran seberapa baik model dalam melakukan prediksi secara keseluruhan dan pada masing-masing kelas.

BAB IV

HASIL DAN ANALISIS

4.1 Hasil Eksplorasi Data (EDA)

Hasil eksplorasi data menunjukkan bahwa terdapat beberapa fitur yang memiliki hubungan dengan performa siswa, seperti kebiasaan belajar dan dukungan keluarga. EDA dilakukan untuk memahami karakteristik data dan membantu proses pemodelan.

4.2 Hasil Training Model

Setelah dilakukan pelatihan menggunakan data latih, model Decision Tree berhasil dibangun dengan struktur pohon yang jelas. Model mampu mempelajari pola hubungan antara fitur dan target klasifikasi performa siswa dengan baik.

4.3 Evaluasi Model

Hasil evaluasi menunjukkan bahwa model Decision Tree memiliki tingkat akurasi yang cukup baik. Nilai precision dan recall menunjukkan bahwa model mampu membedakan siswa dengan performa baik dan kurang baik secara cukup akurat. F1-score digunakan sebagai ukuran keseimbangan antara precision dan recall.

4.4 Visualisasi Pohon Keputusan

Visualisasi pohon keputusan menunjukkan aturan-aturan yang digunakan model dalam melakukan klasifikasi. Setiap node merepresentasikan kondisi tertentu, sedangkan leaf node menunjukkan hasil prediksi performa siswa. Visualisasi ini memudahkan interpretasi hasil model.

4.5 Analisis Hasil

Berdasarkan hasil pengujian, Decision Tree terbukti mampu digunakan untuk memprediksi performa siswa dengan cukup baik. Pembatasan kedalaman pohon membantu mengurangi overfitting, sehingga model dapat melakukan generalisasi pada data uji. Hasil ini menunjukkan bahwa Decision Tree cocok digunakan untuk permasalahan klasifikasi pada bidang pendidikan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa algoritma Decision Tree dapat diterapkan dengan baik untuk memprediksi performa siswa menggunakan Dataset Student Performance. Model mampu mengklasifikasikan performa siswa berdasarkan data akademik dan non-akademik dengan tingkat akurasi yang cukup baik.

Proses preprocessing data dan pemilihan parameter seperti kedalaman pohon (`max_depth`) berpengaruh terhadap hasil model. Dengan pembatasan kedalaman pohon, model menjadi lebih stabil dan tidak terlalu kompleks, sehingga mampu melakukan generalisasi pada data uji.

5.2 Saran

Untuk penelitian selanjutnya, disarankan menggunakan metode ensemble seperti Random Forest atau Gradient Boosting agar performa model dapat ditingkatkan. Selain itu, penggunaan fitur tambahan dan optimasi parameter yang lebih mendalam juga dapat dilakukan untuk memperoleh hasil prediksi yang lebih akurat.

LAMPIRAN

Source Code

```
decision_tree_student.py > y_pred
1  import pandas as pd
2  import matplotlib.pyplot as plt
3
4  from sklearn.model_selection import train_test_split
5  from sklearn.preprocessing import LabelEncoder
6  from sklearn.tree import DecisionTreeClassifier, plot_tree
7  from sklearn.metrics import accuracy_score, classification_report
8
9  # =====
10 # 1. LOAD DATASET
11 # =====
12 data = pd.read_csv("student-mat.csv", sep=';')
13
14 print("Jumlah data:", data.shape)
15 print(data.head())
16
17 # =====
18 # 2. PREPROCESSING DATA
19 # =====
20
21 # Membuat target klasifikasi
22 # Lulus (1) jika nilai akhir >= 10, Tidak lulus (0) jika < 10
23 data['performance'] = data['G3'].apply(lambda x: 1 if x >= 10 else 0)
24
25 # Menghapus kolom nilai asli
26 data.drop(['G1', 'G2', 'G3'], axis=1, inplace=True)
27
28 # Encoding data kategorikal
29 encoder = LabelEncoder()
30 for col in data.columns:
31     if data[col].dtype == 'object':
32         data[col] = encoder.fit_transform(data[col])
33
34 # =====
35 # 3. PEMBAGIAN DATA
36 # =====
37 X = data.drop('performance', axis=1)
38 y = data['performance']
39
40 X_train, X_test, y_train, y_test = train_test_split(
41     X, y, test_size=0.2, random_state=42
42 )
43
44 print("Data training:", X_train.shape)
45 print("Data testing :", X_test.shape)
46
47 # =====
48 # 4. MEMBANGUN MODEL DECISION TREE
49 # =====
50 model = DecisionTreeClassifier(
51     criterion='gini',
52     max_depth=4,
53     random_state=42
54 )
55
56 model.fit(X_train, y_train)
57
58 # =====
59 # 5. EVALUASI MODEL
60 # =====
61 y_pred = model.predict(X_test)
62
63 accuracy = accuracy_score(y_test, y_pred)
64 print("\nAccuracy:", accuracy)
65 print("\nClassification Report:")
66 print(classification_report(y_test, y_pred))
```



```

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

accuracy = accuracy_score(y_test, y_pred)
print("\nAccuracy:", accuracy)
print("\nClassification Report:")
print(classification_report(y_test, y_pred))

# =====
# 6. VISUALISASI POHON KEPUTUSAN
# =====
plt.figure(figsize=(22, 12))
plot_tree(
    model,
    feature_names=X.columns,
    class_names=['Tidak Lulus', 'Lulus'],
    filled=True
)
plt.title("Decision Tree - Student Performance")
plt.show()

```

Link GitHub : <https://github.com/vio-ackerman/UAS-Machine-Learning-DecisionTree.git>

