LAPORAN TUGAS UAS

IMPLEMENTASI ALGORITMA DECESION TREE UNTUK KLASIFIKASI DATASET IRIS

Oleh:

Moh. Fadel Farista 231011401386

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS PAMULANG

2026

1. PENDAHULUAN

1.1 Latar Belakang

Machine learning merupakan salah satu teknologi yang banyak digunakan untuk membantu proses pengambilan keputusan berbasis data. Salah satu algoritma yang sering digunakan dalam permasalahan klasifikasi adalah Decision Tree, karena algoritma ini mudah dipahami dan mampu menjelaskan proses pengambilan keputusan secara visual.

Dataset Iris merupakan dataset klasifikasi yang umum digunakan dalam pembelajaran machine learning. Dataset ini berisi data karakteristik fisik bunga Iris yang dapat digunakan untuk mengklasifikasikan jenis bunga ke dalam tiga kelas. Dengan struktur data yang sederhana dan bersih, dataset ini cocok untuk menguji kinerja algoritma Decision Tree.

Oleh karena itu, penelitian ini bertujuan untuk mengimplementasikan algoritma Decision Tree dalam melakukan klasifikasi pada dataset Iris serta mengevaluasi performa model menggunakan metrik evaluasi yang sesuai.

1.2 Tujuan

Tujuan dari penelitian ini adalah:

1. Mengimplementasikan algoritma Decision Tree untuk melakukan klasifikasi pada dataset Iris
2. Mengevaluasi performa model klasifikasi menggunakan metrik accuracy, precision, recall, dan F1-score
3. Menganalisis proses pengambilan keputusan model melalui visualisasi pohon keputusan

1.3 Rumusan Masalah

1. Bagaimana implementasi algoritma Decision Tree dalam melakukan klasifikasi pada dataset Iris?
2. Bagaimana performa algoritma Decision Tree dalam mengklasifikasikan dataset Iris berdasarkan metrik accuracy, precision, recall, dan F1-score?
3. Bagaimana proses pengambilan keputusan algoritma Decision Tree berdasarkan hasil visualisasi pohon keputusan?

2. LANDASAN TEORI

2.1 Decision Tree

Decision Tree adalah algoritma supervised learning yang dapat digunakan untuk masalah klasifikasi dan regresi. Algoritma ini bekerja dengan cara membagi data secara rekursif berdasarkan fitur-fitur yang paling informatif hingga mencapai kondisi tertentu.

2.2 Komponen Decision Tree

2.2.1 Node

Node adalah titik dalam pohon yang merepresentasikan fitur atau atribut yang digunakan untuk membagi data. Setiap node internal melakukan tes terhadap suatu fitur.

2.2.2 Root

Root adalah node paling atas dalam pohon keputusan. Node ini berisi seluruh dataset dan melakukan splitting pertama berdasarkan fitur yang paling informatif.

2.2.3 Leaf

Leaf (daun) adalah node terminal yang tidak memiliki cabang lagi. Node ini berisi hasil prediksi atau keputusan akhir untuk data yang mencapai node tersebut.

2.2.4 Splitting

Splitting adalah proses membagi node menjadi dua atau lebih sub-node berdasarkan kondisi tertentu dari fitur. Kriteria splitting:

- **Gini Impurity**: Mengukur kemurnian node (0 = murni, 0.5 = tidak murni)
- **Entropy/Information Gain**: Mengukur ketidakpastian atau disorder dalam data

2.2.5 Pruning

Pruning adalah teknik untuk mengurangi kompleksitas pohon dengan memotong cabang yang tidak signifikan. Tujuannya mencegah overfitting dan meningkatkan generalisasi model.

Jenis Pruning:

- **Pre-pruning**: Menghentikan pertumbuhan pohon lebih awal (max_depth, min_samples_split)
- **Post-pruning**: Membangun pohon lengkap lalu memotong cabang yang tidak perlu

2.3 Perbandingan Tree-Based Methods

2.3.1 Decision Tree

- Model tunggal berbentuk pohon keputusan
- Mudah diinterpretasi dan divisualisasikan
- Cepat untuk training dan prediksi
- Rentan terhadap overfitting pada data kompleks

2.3.2 Random Forest

- Ensemble dari banyak decision trees (bagging approach)
- Setiap tree dilatih pada subset data yang berbeda (bootstrap sampling)
- Menggunakan subset fitur random untuk setiap split
- Mengurangi variance dan meningkatkan stabilitas
- Lebih akurat namun kurang interpretable

2.3.3 Gradient Boosting

- Ensemble trees yang dibangun secara sekuensial (boosting approach)
- Setiap tree baru memperbaiki error dari tree sebelumnya
- Mengoptimalkan loss function secara gradual
- Sangat akurat untuk data kompleks
- Memerlukan tuning parameter yang hati-hati

2.4 Kelebihan dan Kekurangan Tree-Based Methods

Kelebihan:

1. **Interpretabilitas**: Mudah dipahami dan divisualisasikan
2. **No Scaling Required**: Tidak memerlukan normalisasi atau standardisasi data
3. **Handle Mixed Data**: Dapat menangani fitur numerik dan kategorikal
4. **Non-linearity**: Mampu menangkap hubungan non-linear
5. **Feature Selection**: Otomatis memberikan feature importance
6. **Robust to Outliers**: Tidak sensitif terhadap outlier

7. **Missing Values**: Beberapa implementasi dapat menangani missing values

Kekurangan:

1. **Overfitting**: Decision tree tunggal mudah overfit
2. **Instability**: Sensitif terhadap perubahan kecil dalam data
3. **Bias**: Cenderung bias terhadap fitur dengan banyak kategori
4. **Extrapolation**: Tidak baik untuk prediksi di luar range training data
5. **Non-smooth Predictions**: Menghasilkan prediksi step-wise
6. **Computational Cost**: Ensemble methods memerlukan resource komputasi besar

3. METODOLOGI

3.1 Dataset

Penelitian ini menggunakan **dataset Iris**, yang merupakan dataset klasifikasi dengan total 150 data. Dataset ini memiliki empat atribut numerik, yaitu *sepal length*, *sepal width*, *petal length*, dan *petal width*, serta satu variabel target berupa kelas bunga Iris yang terdiri dari tiga kelas, yaitu *setosa*, *versicolor*, dan *virginica*. Dataset Iris dipilih karena bersifat sederhana, bersih, dan umum digunakan dalam pembelajaran algoritma klasifikasi.

3.2 Load dan Eksplorasi Data (EDA)

Tahap awal penelitian dilakukan dengan memuat dataset ke dalam lingkungan Python menggunakan library *scikit-learn*. Dataset kemudian dikonversi ke dalam bentuk DataFrame untuk memudahkan analisis. Eksplorasi data dilakukan secara singkat dengan menampilkan beberapa data awal, informasi dataset, serta statistik deskriptif. Tujuan dari tahap ini adalah untuk memahami struktur data, tipe data, serta karakteristik umum dataset sebelum dilakukan pemodelan.

3.3 Data Preprocessing

Pada tahap preprocessing, dilakukan pengecekan terhadap missing value dan tipe data. Hasil pengecekan menunjukkan bahwa dataset Iris tidak memiliki missing value dan seluruh fitur bertipe numerik. Oleh karena itu, tidak diperlukan proses penanganan missing value maupun encoding data kategorikal. Selanjutnya, data dipisahkan menjadi fitur (X) dan target (y) untuk keperluan pemodelan.

3.4 Pembagian Data

Dataset dibagi menjadi dua bagian, yaitu data latih (*training set*) dan data uji (*testing set*). Pembagian data dilakukan dengan perbandingan 80% untuk data latih dan 20% untuk data uji. Data latih digunakan untuk membangun model, sedangkan data uji digunakan untuk mengevaluasi performa model yang dihasilkan.

3.5 Pembangunan Model Decision Tree

Model klasifikasi dibangun menggunakan algoritma Decision Tree dengan bantuan library *scikit-learn*. Pada tahap ini ditentukan beberapa parameter penting, yaitu kriteria pemisahan (*criterion*) menggunakan metode *gini* dan kedalaman maksimum pohon (*max_depth*) untuk menghindari terjadinya overfitting. Model kemudian dilatih menggunakan data latih yang telah disiapkan.

3.6 Evaluasi Model

Evaluasi model dilakukan menggunakan data uji untuk mengetahui performa klasifikasi yang dihasilkan. Karena penelitian ini merupakan kasus klasifikasi, metrik evaluasi yang digunakan meliputi accuracy, precision, recall, dan F1-score. Metrik-metrik tersebut digunakan untuk mengukur tingkat ketepatan dan kualitas model dalam mengklasifikasikan data ke dalam kelas yang benar.

3.7 Visualisasi Pohon Keputusan

Sebagai tahap akhir, struktur pohon keputusan divisualisasikan untuk melihat proses pengambilan keputusan yang dilakukan oleh model. Visualisasi ini membantu dalam memahami fitur yang paling berpengaruh serta alur pemisahan data dari root hingga leaf node. Hasil visualisasi juga digunakan sebagai pendukung analisis dalam penelitian ini.

4. HASIL DAN ANALISIS

4.1 Hasil Exploratory Data Analysis (EDA)

Berdasarkan hasil eksplorasi awal terhadap dataset Iris, diketahui bahwa dataset terdiri dari 150 data dengan 4 fitur numerik, yaitu *sepal length*, *sepal width*, *petal length*, dan *petal width*, serta 3 kelas target (*setosa*, *versicolor*, dan *virginica*). Hasil pengecekan menunjukkan bahwa dataset tidak memiliki missing value, sehingga data dapat langsung digunakan untuk pemodelan.

Statistik deskriptif menunjukkan bahwa fitur *petal length* dan *petal width* memiliki variasi nilai yang lebih besar dibandingkan fitur sepal. Hal ini mengindikasikan bahwa fitur petal memiliki peran penting dalam membedakan kelas bunga Iris. Visualisasi *pairplot* juga memperlihatkan bahwa kelas *setosa* terpisah dengan jelas dari dua kelas lainnya, sedangkan kelas *versicolor* dan *virginica* memiliki beberapa area yang saling tumpang tindih.

4.2 Hasil Pembangunan Model Decision Tree

Model Decision Tree dibangun menggunakan parameter *criterion* = gini dan *max_depth* = 3. Pembatasan kedalaman pohon dilakukan untuk mencegah model menjadi terlalu kompleks dan mengurangi risiko overfitting. Model dilatih menggunakan 80% data latih dan diuji pada 20% data uji

Hasil pengujian menunjukkan bahwa model mampu melakukan klasifikasi dengan akurasi yang tinggi. Nilai precision, recall, dan F1-score pada masing-masing kelas juga menunjukkan hasil yang baik dan relatif seimbang. Hal ini menandakan bahwa model tidak hanya akurat secara keseluruhan, tetapi juga konsisten dalam memprediksi setiap kelas

4.3 Analisis Confusion Matrix

Berdasarkan confusion matrix yang dihasilkan, sebagian besar data uji berhasil diklasifikasikan dengan benar. Kelas *setosa* umumnya dapat diprediksi dengan sangat baik tanpa kesalahan, karena karakteristiknya yang cukup berbeda dari kelas lain. Kesalahan klasifikasi yang terjadi sebagian besar terdapat pada kelas *versicolor* dan *virginica*, yang memang memiliki karakteristik fitur yang saling mendekati.

Tidak ditemukan kesalahan klasifikasi yang dominan pada satu kelas tertentu, sehingga dapat disimpulkan bahwa model tidak mengalami bias. Confusion matrix ini memperkuat hasil evaluasi metrik klasifikasi yang menunjukkan performa model yang stabil

4.4 Visualisasi dan Interpretasi Decision Tree

Hasil visualisasi pohon keputusan menunjukkan struktur pohon yang relatif sederhana dan mudah dipahami. Setiap node merepresentasikan proses pemisahan data berdasarkan nilai fitur tertentu, sementara node daun menunjukkan hasil prediksi kelas. Dengan *max_depth* = 3, jumlah cabang pada pohon tetap terbatas sehingga model tetap interpretable.

Dari visualisasi tersebut terlihat bahwa fitur *petal length* dan *petal width* menjadi fitur utama yang digunakan dalam proses pemisahan data. Hal ini sejalan dengan hasil EDA yang menunjukkan bahwa fitur petal memiliki variasi dan daya diskriminasi yang lebih tinggi dibandingkan fitur sepal

4.5 Feature Importance

Analisis *feature importance* menunjukkan bahwa kontribusi terbesar dalam pengambilan keputusan berasal dari fitur *petal length* dan *petal width*. Fitur *sepal length* dan *sepal width* memiliki pengaruh yang lebih kecil. Temuan ini memperkuat hasil eksplorasi data dan visualisasi pohon keputusan bahwa fitur petal merupakan faktor utama dalam klasifikasi bunga Iris.

4.6 Insight Keseluruhan

Secara keseluruhan, algoritma Decision Tree dengan parameter yang digunakan mampu memberikan performa klasifikasi yang sangat baik pada dataset Iris. Model tidak hanya akurat, tetapi juga mudah dipahami dan diinterpretasikan. Hal ini menjadikan Decision Tree sebagai metode yang efektif untuk studi kasus klasifikasi dengan dataset berukuran kecil hingga menengah seperti Iris.

5. KESIMPULAN

5.1 Kesimpulan Umum

Berdasarkan hasil penelitian yang telah dilakukan, algoritma Decision Tree mampu mengklasifikasikan dataset Iris dengan performa yang baik. Model berhasil mempelajari pola dari data latih dan memberikan hasil prediksi yang akurat pada data uji. Hasil evaluasi menggunakan metrik klasifikasi menunjukkan bahwa Decision Tree merupakan metode yang efektif untuk menyelesaikan permasalahan klasifikasi pada dataset Iris.

5.2. Faktor yang Mempengaruhi Performa Model

Beberapa faktor yang mempengaruhi performa model Decision Tree pada penelitian ini antara lain pemilihan parameter model seperti *max_depth* dan *criterion*. Pembatasan kedalaman pohon membantu mencegah terjadinya overfitting dan meningkatkan kemampuan generalisasi model. Selain itu, kualitas fitur pada dataset juga berperan penting, di mana fitur *petal length* dan *petal width* memiliki kontribusi besar dalam meningkatkan akurasi klasifikasi.

5.3 Kelebihan Tree-Based Methods pada Studi Kasus

Pada studi kasus dataset Iris, metode berbasis pohon keputusan memiliki kelebihan utama berupa kemudahan interpretasi dan visualisasi. Struktur pohon keputusan memungkinkan pengguna untuk memahami proses pengambilan keputusan model secara jelas. Selain itu, Decision Tree tidak memerlukan preprocessing yang kompleks, seperti normalisasi atau encoding, sehingga lebih sederhana dalam penerapannya.

5.4 Kesimpulan Akhir

Secara keseluruhan, algoritma Decision Tree terbukti mampu memberikan hasil klasifikasi yang baik dan mudah dipahami pada dataset Iris. Dengan parameter yang tepat, model dapat menghasilkan performa yang optimal tanpa kompleksitas yang berlebihan. Oleh karena itu, Decision Tree sangat sesuai digunakan sebagai metode pembelajaran dan studi kasus dalam penerapan algoritma klasifikasi berbasis *machine learning*.

Link github

https://github.com/mohfadelfarista/uas_mohfadelfarista