

IMPLEMENTASI BIJI LABU UNTUK KLASIFIKASI MENGGUNAKAN ALGORITMA DECISION TREE

Rayhan Rizal Mahendra¹

¹Universitas Pembangunan Nasional “Veteran” Jawa Timur
20081010045@student.upnjatim.ac.id

ABSTRACT

This paper gives complete guidelines for authors submitting papers for the AIRCC Journals.

KEYWORDS

Network Protocols, Wireless Network, Mobile Network, Virus, Worms & Trojans

1. INTRODUCTION

Data Mining merupakan serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual. Sehingga Data Mining menjadi alat yang semakin penting untuk mengubah data tersebut menjadi informasi [1]. Data Mining dan Pohon Keputusan (Decision Tree) merupakan dua konsep yang saling terkait dalam dunia analisis data dan pengambilan keputusan. Dalam era informasi yang semakin berkembang pesat, jumlah data yang dihasilkan setiap harinya semakin melonjak. Oleh karena itu, penting bagi kita untuk memiliki metode yang efektif dalam mengeksplorasi dan menggali informasi berharga dari kumpulan data yang besar dan kompleks.

Data Mining, atau pertambangan data, merujuk pada proses identifikasi pola atau penemuan informasi yang signifikan dari suatu dataset. Metode ini melibatkan sejumlah teknik analisis statistik, matematika, dan kecerdasan buatan untuk mengungkap hubungan, kecenderungan, dan pola tersembunyi yang mungkin tidak terlihat secara langsung. Data Mining memainkan peran penting dalam mendukung pengambilan keputusan dengan memberikan wawasan mendalam tentang data yang dimiliki.

Salah satu alat yang efektif dalam implementasi Data Mining adalah decision tree. Decision tree merupakan model representasi grafis dari keputusan dan konsekuensinya. Metode ini menggunakan struktur pohon dengan node sebagai keputusan atau keadaan dan cabang sebagai hasil dari keputusan tersebut. Untuk dataset yang digunakan adalah biji Labu yang diambil dari Kaggle sebagai bahan uji klasifikasi dengan Decision Tree. Penelitian ini diharapkan mampu memberikan hasil klasifikasi berupa decision tree yang dapat digunakan sebagai bahan pertimbangan dalam menentukan dua jenis labu.

2. TINJAUAN PUSTAKA

2.1. Decision Tree

Pohon Keputusan (Decision Tree) adalah suatu model prediktif dalam analisis data yang digunakan untuk memetakan keputusan dan kemungkinan konsekuensinya dalam bentuk struktur pohon. Model ini memiliki tampilan grafis seperti pohon dengan cabang-cabang yang merepresentasikan keputusan atau keadaan, dan daun-daun yang mewakili hasil atau konsekuensi. Pada setiap simpul keputusan (node), dilakukan pemilihan opsi berdasarkan fitur atau atribut

tertentu, dan proses ini terus berlanjut hingga mencapai daun pohon yang menunjukkan hasil akhir atau prediksi. Pohon (tree) adalah sebuah struktur data yang terdiri dari simpul (node) dan rusuk (edge). Simpul pada sebuah pohon dibedakan menjadi tiga, yaitu simpul akar (root/node), simpul percabangan/internal (branch/internal node) dan simpul daun (leaf node) [2].

Keuntungan utama dari Pohon Keputusan melibatkan kemampuan interpretasi yang tinggi, karena model ini menyajikan keputusan dan hasil secara hierarkis dan terstruktur. Selain itu, Pohon Keputusan dapat menangani baik masalah klasifikasi (pemisahan ke dalam kategori atau kelas) maupun regresi (prediksi nilai berkelanjutan). Model ini juga dapat mengatasi data yang tidak seimbang dengan baik.

Meskipun memiliki banyak kelebihan, Pohon Keputusan juga memiliki beberapa kelemahan, seperti kecenderungan overfitting (terlalu sesuai dengan data pelatihan) dan sensitivitas terhadap perubahan kecil dalam data. Oleh karena itu, pemilihan algoritma dan parameter yang tepat, serta pengelolaan overfitting, menjadi kunci penting dalam mengimplementasikan Pohon Keputusan dengan efektif.

2.2. Klasifikasi

Klasifikasi adalah proses dari mencari suatu himpunan model (fungsi) yang dapat mendeskripsikan dan membedakan kelas-kelas data atau konsep-konsep, dengan tujuan dapat menggunakan model tersebut untuk memprediksi kelas dari suatu objek yang mana kelasnya belum diketahui [3]. Klasifikasi seringkali melibatkan penggunaan algoritma atau aturan yang diterapkan pada data atau objek tertentu untuk menentukan kategori di mana mereka harus ditempatkan. Dalam konteks kecerdasan buatan, teknik klasifikasi sering digunakan untuk pengembangan model prediktif, seperti Pohon Keputusan, K-Nearest Neighbors, dan Jaringan Saraf Tiruan, untuk mencapai pengenalan pola dan pengambilan keputusan otomatis.

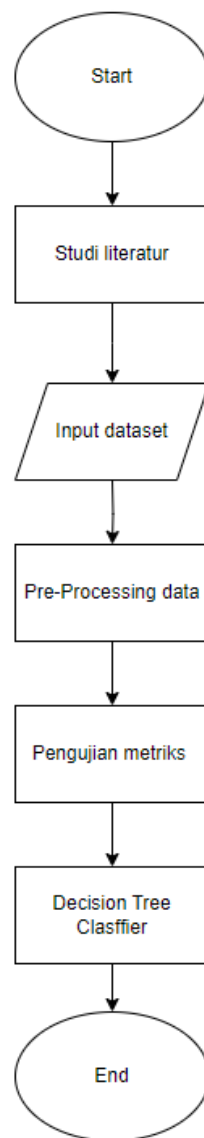
2.3. Python

Python adalah bahasa pemrograman tingkat tinggi yang bersifat serba guna, mudah dipahami, dan memiliki sintaksis yang bersih. Diciptakan oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991, Python dirancang untuk mempromosikan produktivitas dan membaca kode dengan mudah. Sejak itu, Python telah menjadi salah satu bahasa pemrograman paling populer di dunia dan digunakan di berbagai bidang, termasuk pengembangan perangkat lunak, ilmu data, kecerdasan buatan, pengembangan web, dan pemrograman sistem.

Python digunakan secara luas di berbagai industri dan disukai oleh pengembang untuk proyek-proyek berbagai ukuran. Oleh karena itu, Python telah menjadi salah satu bahasa pemrograman yang paling penting dan serbaguna dalam dunia teknologi informasi.

3. METODE PENELITIAN

Pada tahap ini, terdapat beberapa tahapan dalam melakukan metode penelitian untuk klasifikasi biji labu menggunakan algoritma Decision Tree sebagai berikut :



Gambar 1. Flowchart

1. Studi Literatur

Tahap studi literatur ini merupakan langkah penelitian untuk mengumpulkan dasar teori-teori yang diperlukan dalam penelitian [4]. Studi literatur ini sangat bermanfaat untuk menambah pengetahuan melalui dasar teori sebelum penelitian dilakukan.

2. Input dataset

Untuk dataset yang penulis gunakan adalah dataset biji labu yang diambil dari situs Kaggle. Dataset ini diuji penelitian untuk mengetahui kualitas dari biji labu.

3. Pre-Processing Data

Data preprocessing adalah proses transformasi, menggabungkan, atau mengubah data menjadi bentuk yang sesuai, agar dapat diproses dengan perhitungan algoritma decision

tree [5]. Data penelitian yang diperoleh merupakan data mentah (data asli atau primer) yang perlu dilakukan preprocessing. Tahap ini termasuk menentukan atribut yang akan digunakan dalam proses klasifikasi dan melakukan modifikasi pada data dengan menangani data yang hilang, data ganda, dan mengubah data atribut menjadi tipe kategori.

4. Pengujian Metriks

Pengujian metrics yang penulis gunakan adalah pengujian akurasi skor pada biji labu. Pengujian ini menggunakan variable X_train dan y_train sebagai fitur dan target pada kolom dataset. Akurasi adalah salah satu metrik yang umum digunakan untuk mengukur seberapa baik model klasifikasi berkinerja dalam memprediksi kelas atau label yang benar. Namun, penting untuk menyadari bahwa akurasi sendiri mungkin tidak selalu mencerminkan kualitas keseluruhan dari suatu model, terutama jika dataset tidak seimbang atau ada lebih banyak kelas dari yang lain.


5. Decision Tree Classifier

Tahapan terakhir adalah implementasi dari Decision Tree Classifier. Decision Tree Classifier (Klasifikasi Pohon Keputusan) adalah algoritma pembelajaran mesin yang digunakan untuk melakukan klasifikasi berdasarkan serangkaian keputusan yang diambil dari fitur-fitur dalam data. Model ini memodelkan keputusan berdasarkan hierarki pohon, di mana setiap simpul dalam pohon mewakili suatu keputusan atau pengujian pada suatu fitur. Algoritma ini secara berulang membagi data menjadi subset yang lebih kecil berdasarkan fitur-fitur tertentu hingga mencapai simpul daun yang menunjukkan kelas atau label akhir.

4. HASIL DAN PEMBAHASAN

1. Pengumpulan data

Untuk dataset yang dipakai adalah dataset biji labu yang diambil dari situs Kaggle.



	Area	Perimeter	Major_Axis_Length	Minor_Axis_Length	Convex_Area	Equiv_Diameter	Eccentricity	Solidity	Extent	Roundness	Aspect_Ration	Compactness	Class
0	56276	888.242	326.1485	220.2388	56831	267.6805	0.7376	0.9902	0.7453	0.8963	1.4809	0.8207	Çerçevelek
1	76631	1068.146	417.1932	234.2289	77280	312.3614	0.8275	0.9916	0.7151	0.8440	1.7811	0.7487	Çerçevelek
2	71623	1082.987	435.8328	211.0457	72663	301.9822	0.8749	0.9857	0.7400	0.7674	2.0651	0.6929	Çerçevelek
3	66458	992.051	381.5638	222.5322	67118	290.8899	0.8123	0.9902	0.7396	0.8486	1.7146	0.7624	Çerçevelek
4	66107	998.146	383.8883	220.4545	67117	290.1207	0.8187	0.9850	0.6752	0.8338	1.7413	0.7557	Çerçevelek

Gambar 2. Dataset

2. Pre-Processing data

Tahapan ini menghasilkan beberapa uji dataset seperti pengecekan kolom apakah ada yang kosong atau tidak.

```
[ ] nan_in_columns = df_cleaned.isna().any()

# Display columns with NaN values
print("Columns with NaN values:")
print(nan_in_columns[nan_in_columns].index)
```

```
Columns with NaN values:
Index([], dtype='object')
```

Gambar 3. Columns with Nan Values

Bisa dilihat pada gambar 3, untuk kolom dengan Nan values tidak ada. Maka bisa dilanjutkan untuk tahap berikutnya.

Memisahkan antara fitur dan target. Untuk fitur ini terdapat pada dataset yaitu semua kolom kecuali class. Class sendiri akan masuk kedalam target. Tahap selanjutnya adalah merubah kolom Class menjadi numerik dengan melakukan label encoder.

```
[ ] le = LabelEncoder()
    df_cleaned_copy = df_cleaned.copy()
    df_cleaned_copy['Class'] = le.fit_transform(df_cleaned_copy['Class'])
```

Gambar 4. Label encoder

Setelah kolom Class diubah menjadi numerik, Kemudian menjadikan semua fitur dan target menjadi array agar mudah untuk diuji selanjutnya.

```
[[5.627600e+04 8.882420e+02 3.261485e+02 ... 1.480900e+00 8.207000e-01
 0.000000e+00]
 [7.663100e+04 1.068146e+03 4.171932e+02 ... 1.781100e+00 7.487000e-01
 0.000000e+00]
 [7.162300e+04 1.082987e+03 4.358328e+02 ... 2.065100e+00 6.929000e-01
 0.000000e+00]
 ...
 [8.799400e+04 1.210314e+03 5.072200e+02 ... 2.282800e+00 6.599000e-01
 1.000000e+00]
 [8.001100e+04 1.182947e+03 5.019065e+02 ... 2.451300e+00 6.359000e-01
 1.000000e+00]
 [8.493400e+04 1.159933e+03 4.628951e+02 ... 1.973500e+00 7.104000e-01
 1.000000e+00]]
```

Gambar 5. Hasil array

Kemudian pada tahap pre processing dilakukan split dataset. Menggunakan library train test split. Setelah itu melakukan standarisasi menggunakan scaler berdasarkan X_train dan X_test.

```
[ ] from sklearn.model_selection import train_test_split

    # Split dataset menjadi training set dan test set
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=39)
```

Gambar 6. Split dataset

Tahapan selanjutnya adalah standarisasi pada X_train dan X_test. Tujuannya adalah agar data uji training dan test dapat mengoptimalkan performa model yang akan dibuat, mengurangi outlier dan mempermudah interpretasi.

```
scaler = StandardScaler()

X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Gambar 7. Standarisasi

3. Pengujian metrics

Untuk pengujian metrics yang dilakukan adalah pengujian metrics accuracy score. Didapatkan train accuracy sebesar 96.20% dan test accuracy 87.20%

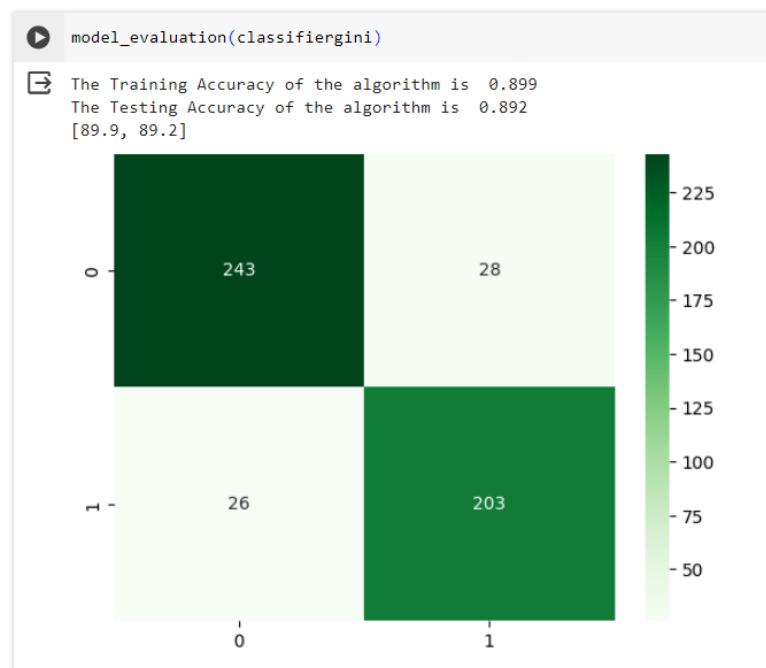
```
➡ Train accuracy: 96.20 %  
Test accuracy: 87.20 %
```

Gambar 8 Pengujian metrics

4. Implementasi Decision Tree Classifier

Pada proses klasifikasi terdapat library yang harus digunakan pada pengujian ini dengan menggunakan scikit-learn yang digunakan dalam pemograman machine-learning. Sebelum implementasi dilakukan pre-processing yang telah disebutkan pada halaman atas. Pada implementasi algoritma ini dilakukan X_train dan y_train sebagai classifiernya dan classifier ent. Pada decision tree ini akan membuat pohon keputusan berdasarkan fitur – fitur data untuk memprediksi class yaitu Cercevelik dan Urgup_Sivrisi.

Didapatkan untuk model evaluation dengan classifiernya mendapatkan training akurasi sebesar 89,9. Dan untuk testing akurasi sebesar 89,2.



Gambar 9. Classifiernya

Keterangan :

True Negative: 243

True Positive: 203

False Positive: 28

False Negative: 26

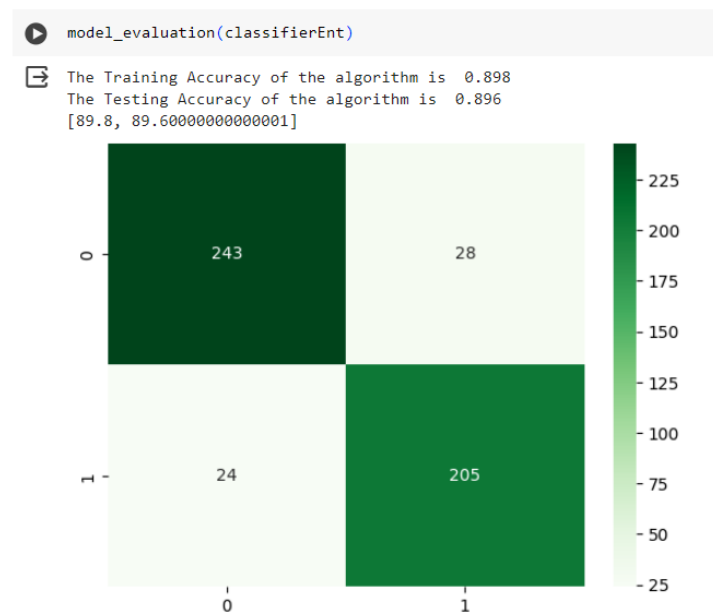
TN: jumlah data yang berhasil diprediksi nilainya 0

TP: jumlah data yang berhasil di prediksi 1

FP: jumlah data yang diprediksi 1, tapi data aslinya 0

FN: jumlah data sebenarnya positif, tetapi di prediksi negatif

Untuk ClassifierEnt ini didapatkan training akurasi sebesar 89,8 dan untuk testing akurasi sebesar 89,6.



Gambar 10. ClassifierEnt

Keterangan :

True Negative: 243

True Positive: 205

False Positive: 28

False Negative: 24

TN: jumlah data yang berhasil diprediksi nilainya 0

TP: jumlah data yang berhasil di prediksi 1

FP: jumlah data yang diprediksi 1, tapi data aslinya 0

FN: jumlah data sebenarnya positif, tetapi di prediksi negatif

5. KESIMPULAN

Pada penelitian ini, penggunaan atribut pada klasifikasi pohon keputusan untuk dataset biji Labu. Analisis dilakukan dengan beberapa tahapan yaitu input dataset, preprocessing, pengujian metrics dan implementasi decision tree.

Hasil evaluasi kinerja decision tree mencapai hasil tertinggi dalam classifierni mendapatkan training dan testing accuracy sebesar 89,9% dan 89,2%. Selanjutnya pada classifierEnt mendapatkan traing dan testing accuracy sebesar 89,8% dan 89,6%.

REFERENCES

- [1] B. Utami and P. Aliandu, "KLASIFIKASI PENENTUAN TIM UTAMA OLAHRAGA HOCKEY MENGGUNAKAN ALGORITMA C4.pdf," *Proc. Int. Conf. Information, Commun. Technol. Syst.*, vol. 5, no. 4, pp. 1–5, 2013.
- [2] J. Eska, "Data Mining Untuk Prediksi Penjualan Wallpaper Menggunakan Algoritma C45," *JURTEKSI (Jurnal Teknol. dan Sist. Informasi)*, vol. 2, pp. 9–13, 2016.
- [3] D. F. Ristianti, "Komparasi Algoritma Klasifikasi pada Data Mining," vol. 1, no. 1, pp. 148–156, 2019.
- [4] Suryani, D. Rahmadani, A. A. Muzafar, A. Hamid, R. Annisa, and Mustakim, "Analisis Perbandingan Algoritma C4.5 dan CART untuk Klasifikasi Penyakit Stroke," in *SENTIMAS: Seminar Nasional Penelitian dan Pengabdian Masyarakat*, 2022, pp. 197–206.
- [5] A. A. Aldino and H. Sulistiani, "Decision Tree C4.5 Algorithm for Tuition Aid Grant Program Classification (Case Study: Department of Information System, Universitas Teknokrat Indonesia)," *Educic - Scientific Journal of Informatics Education*, vol. 7, no. 1, pp. 40–50, 2020.