Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) p-ISSN: 2355-7699
Vol. 12, No. 4, Agustus 2025, hlm. 875-884 e-ISSN: 2528-6579

ANALISIS PERBANDINGAN MODEL *MACHINE LEARNING TREE-BASED* DAN *NON-TREE-BASED U*NTUK TUGAS KLASIFIKASI

Fadhilah Hilmi[1], Kenzie Taqiyassar[2], Naufal Romero Putra Pratama[3], Satrio Condro Kusuma[4], Hafiz Rizky Nurwachid[5], Tirana Noor Fatyanosa[6]

[1,2,3,4,5,6,7]Universitas Brawijaya, Malang
Email: [1]fadhilahhilmi@student.ub.ac.id, [2]kenzie19@student.ub.ac.id, [3]naufalromero@student.ub.ac.id, [4]satriocondro@student.ub.ac.id, [5]hafizrizky@student.ub.ac.id, [6]fatyanosa@ub.ac.id
*Penulis Korespondensi

(Naskah masuk: 27 November 2024, diterima untuk diterbitkan: 27 Agustus 2025)

Abstrak

Penelitian ini membahas perbandingan performa model *machine learning* berbasis pohon keputusan (*Tree-Based*) dan non-pohon keputusan (*Non-Tree-Based*) dalam tugas klasifikasi. Model *Tree-based* yang diuji meliputi LightGBM, CatBoost, XGBoost, dan Random Forest, sedangkan model *Non-tree-based* meliputi SVM, KNN, dan GaussianNB. Evaluasi dilakukan pada tiga *dataset* berbeda, yaitu *Spaceship Titanic*, *Horse Health*, dan *Keep It Dry*. Metrik yang digunakan untuk mengevaluasi performa model adalah AUC-ROC, akurasi, dan F1-score Micro. Hasil penelitian menunjukkan bahwa model berbasis pohon keputusan seperti CatBoost dan LightGBM umumnya memberikan performa yang lebih baik dibandingkan dengan model non-pohon keputusan. CatBoost khususnya menunjukkan hasil terbaik dalam hal akurasi, AUC-ROC, dan F1-score Micro di sebagian besar *dataset* yang diuji. Selain itu, penelitian ini juga menyoroti pentingnya pemilihan model yang tepat berdasarkan karakteristik *dataset* yang digunakan. Faktor-faktor seperti kompleksitas data, jumlah fitur, dan distribusi kelas sangat mempengaruhi hasil akhir dari setiap model yang diterapkan. Dengan demikian, temuan ini dapat membantu praktisi *machine learning* dalam memilih model yang paling sesuai untuk tugas klasifikasi tertentu.

Kata kunci: *Machine Learning, Model Tree-Based, Model Non-Tree-Based, Klasifikasi*

COMPARATIVE ANALYSIS OF TREE-BASED AND NON-TRE-BASED MACHINE LEARNING MODELS FOR CLASSIFICATION TASKS

Abstract

This study discusses the performance comparison of tree-based and non-tree-based machine learning models for classification tasks. The Tree-based models tested include LightGBM, CatBoost, XGBoost, and Random Forest, while the Non-tree-based models include SVM, KNN, and GaussianNB. The evaluation was conducted on three different datasets, namely Spaceship Titanic, Horse Health, and Keep It Dry. The metrics used to evaluate model performance are AUC-ROC, accuracy, and F1-score Micro. The results show that tree-based models such as CatBoost and LightGBM generally provide better performance compared to non-tree-based models. CatBoost, in particular, showed the best results in terms of accuracy, AUC-ROC, and F1-score Micro in most of the datasets tested. Additionally, this study highlights the importance of selecting the appropriate model based on the characteristics of the datasets used. Factors such as data complexity, number of features, and class distribution significantly affect the final results of each applied model. Thus, these findings can assist machine learning practitioners in choosing the most suitable model for specific classification tasks.

Keywords: *Machine Learning, Tree-Based Model, Non-Tree-Based Model, Classification*

1. PENDAHULUAN

Machine Learning (ML) atau Pembelajaran Mesin adalah cabang dari kecerdasan buatan yang berfokus pada pengembangan algoritma dan model yang memungkinkan sistem komputer untuk belajar dari dan membuat keputusan berdasarkan data. Inti dari pembelajaran mesin adalah kemampuannya untuk mengidentifikasi pola dan wawasan dari data yang diberikan, serta memperbaiki kinerjanya seiring dengan bertambahnya data dan pengalaman. Berbeda dengan pendekatan tradisional yang memerlukan pemrograman manual untuk setiap tugas spesifik, pembelajaran mesin memungkinkan sistem untuk secara otomatis menyesuaikan diri dan memperbaiki kinerjanya tanpa harus diprogram ulang secara terus-menerus (Alpayadin, 2020). Dalam beberapa dekade terakhir, *machine learning* telah menjadi alat yang

sangat penting dalam berbagai bidang, mulai dari pengenalan pola hingga pengambilan keputusan otomatis (Jayatilake dan Ganegoda, 2021). Model-model *machine learning* telah digunakan untuk berbagai tugas klasifikasi, seperti deteksi spam, diagnosis medis, dan analisis sentimen (Fatyanosa dan Bachtiar, 2017). Dua pendekatan utama dalam machine learning untuk tugas klasifikasi adalah model berbasis pohon keputusan (*Tree-based*) dan non-pohon keputusan (*Non-tree-based*).

Model *Tree-based*, seperti Random Forest dan *Gradient Boosting Machines* (termasuk LightGBM dan XGBoost), merupakan metode *ensemble* yang dibangun berdasarkan *Decision Tree*. Model-model ini telah mendapatkan popularitas yang signifikan karena kemampuannya untuk menangani data dengan fitur yang kompleks dan interaksi non-linear (Jiang dkk, 2020). Decision Tree bekerja dengan membagi data secara rekursif berdasarkan fitur-fitur yang paling informatif, sehingga mudah diinterpretasikan dan diimplementasikan.

Di sisi lain, model *Non-tree-based*, seperti Support Vector Machines (SVM), Naive Bayes (NB), dan K-Nearest Neighbors (KNN), menawarkan pendekatan yang berbeda dalam memodelkan data. SVM menggunakan *hyperplane* untuk memisahkan kelas-kelas dalam data, Naive Bayes mengandalkan teorema probabilitas Bayes, dan KNN mengklasifikasikan data berdasarkan kedekatan dengan data latih.

Terdapat penelitian serupa sebelumnya yang bertujuan membandingkan efektivitas model *ensemble machine learning tree-based* dalam meramalkan arah pergerakan harga saham (Ampomah dkk, 2020). Model yang dibandingkan meliputi Random Forest, XGBoost Classifier, Bagging Classifier, AdaBoost Classifier, Extra Trees Classifier, dan Voting Classifier. *dataset* yang digunakan adalah delapan *dataset* saham dari tiga bursa (NYSE, NASDAQ, dan NSE) yang dikumpulkan secara acak. Setiap *dataset* dibagi menjadi set pelatihan dan set pengujian. Akurasi dari 10-*fold cross validation* digunakan untuk mengevaluasi model pada set pelatihan. Selain itu, model *machine learning* dievaluasi pada set pengujian menggunakan akurasi, presisi, *recall*, F1-score, spesifisitas, dan *area under receiver operating characteristics curve* (AUC-ROC). Uji konkordansi Kendall W digunakan untuk memeringkat kinerja algoritma berbasis pohon. Untuk set pelatihan, model AdaBoost menunjukkan kinerja terbaik. Untuk set pengujian, metrik akurasi, presisi, F1-score, dan AUC menunjukkan hasil signifikan, dengan *Extra Trees classifier* mengungguli model lainnya (Ampomah dkk, 2020).

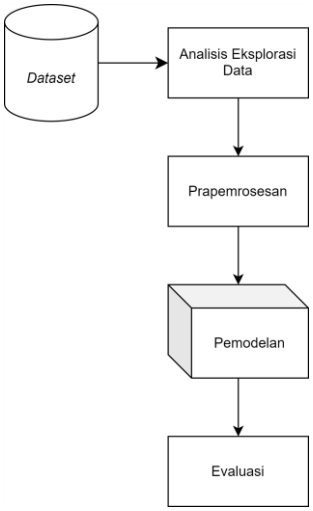
Tujuan dari penelitian ini adalah untuk mencari model terbaik berdasarkan kondisi *dataset* yang ada. Dengan mengevaluasi performa berbagai model *machine learning* pada beberapa *dataset* yang berbeda, penelitian ini bertujuan untuk mengidentifikasi model yang paling efektif dan efisien dalam menangani tugas klasifikasi tertentu. Selain itu, penelitian ini juga berupaya memberikan panduan bagi praktisi *machine learning* dalam memilih model yang paling sesuai berdasarkan karakteristik spesifik dari *dataset* yang digunakan.

2. METODE PENELITIAN

Penelitian ini menggunakan tiga *dataset* berbeda yang diambil dari situs *kaggle.com*. Penggunaan berbagai *dataset* tersebut bertujuan untuk menguji performa model dalam berbagai skenario data. Skenario-skenario tersebut mencakup:

1. Performa model terhadap karakteristik atau distribusi data yang berbeda.
2. Performa model terhadap data yang tidak lengkap.
3. Performa model terhadap data dengan variasi jumlah baris dan kolom atau fitur yang berbeda.

Alur penelitian secara keseluruhan dapat dilihat pada Gambar 1.



Gambar 1. Alur Penelitian

Tabel 1. Ringkasan Informasi *dataset*

dataset	**Jenis Klasifikasi**	**Dimensi**	**Jumlah Fitur numerik & kategori**
Spaceship Titanic	Biner	(8693, 13)	(6, 7)
Predict Health Outcome of Horse	Multikelas	(26570, 25)	(20, 5)
Predict Failures Keep it Dry	Biner	(1235, 29)	(11, 18)

2.1 *dataset*

Tabel 1 menunjukkan informasi umum mengenai setiap *dataset* yang digunakan, nama *dataset*, Jenis Klasifikasi, Ukuran Baris dan Kolom,

serta jumlah Fitur numerik dan kategori. Adapun *dataset* yang digunakan bersumber dari *kaggle.com*.

2.2 Analisis Eksplorasi Data

Pada tahap ini, penekanan dilakukan untuk mencari pengetahuan yang lebih mendalam terkait *dataset* melalui beberapa hal seperti memeriksa keutuhan data, memeriksa distribusi data baik data numerik, kategori, maupun data kelas, serta bersamaan dengan itu, dilakukan deteksi data *outliers* atau pencilan.

Analisis distribusi data numerik divisualisasikan menggunakan plot Kernel Density Estimation (KDE). Plot ini membantu memberikan visualisasi distribusi data numerik secara halus dengan mengubah data diskrit menjadi data kontinu (Whig dkk, 2022). Lalu, untuk analisis distribusi data kategori dan kelas divisualisasikan dengan *Pie Chart*. Terakhir, untuk melakukan deteksi pencilan dilakukan dengan visualisasi *Boxplot*.

2.3 Prapemrosesan

Dalam penelitian ini, penekanan diberikan pada standarisasi tahapan prapemrosesan untuk semua *dataset* yang dianalisis. Untuk mengatasi nilai yang hilang pada data numerik, digunakan imputasi dengan nilai rata-rata, diikuti dengan normalisasi menggunakan metode *Min-Max Scaling*. Pendekatan ini bertujuan untuk memastikan konsistensi skala setiap kolom, sehingga tidak membebani kinerja model. Sementara untuk data kategori yang memiliki nilai yang hilang, digunakan imputasi dengan nilai modus, dan kemudian menerapkan *encoding* menggunakan metode *one-hot encoding*.

Namun, ditemukan beberapa kasus adanya informasi tersembunyi dalam salah satu kolom pada data *Spaceship* Titanic dan *Predict Failure Keep it Dry*. Oleh karena itu, diterapkan *Feature Engineering* pada kedua *dataset* tersebut. Tujuan dari langkah ini adalah untuk meningkatkan performa model dengan mengidentifikasi dan memanfaatkan informasi yang tersembunyi dalam kolom tersebut.

2.4 Pemodelan

Pada tahap pemodelan, terdapat tujuh model yang digunakan, di antaranya *Random Forest*, LightGBM, CatBoost, XGBoost, SVM, KNN, dan Naive Bayes.

2.4.1 *Random Forest*

Random Forest adalah salah satu algoritma *machine learning* yang menggunakan teknik *ensemble*, dengan tujuan untuk membuat lebih dari satu pohon keputusan yang mana ini bisa disebut sebagai *bagging*. Selanjutnya hasil prediksi dari tiap pohon keputusan akan digabungkan untuk mengambil mayoritas suara sebagai hasil akhir (Genuer dan Poggi, 2020).

2.4.2 *XGBoost*

Extreme Gradient Boosting (XGBoost) adalah salah satu algoritma *machine learning* yang termasuk dalam algoritma *Tree-Based* namun di sisi lain juga termasuk ke dalam keluarga algoritma *boosting*. Menggunakan teknik gradient *boosting*, XGBoost membangun model prediksi yang kuat dengan menggabungkan beberapa pohon keputusan yang lemah secara bertahap, sehingga setiap pohon baru berusaha memperbaiki kesalahan dari pohon sebelumnya. Dikenal dengan kemampuannya menangani missing values, menyediakan regularisasi L1 dan L2 untuk mengurangi *overfitting*, serta mendukung komputasi paralel, XGBoost menjadi pilihan populer dalam kompetisi *data science* dan aplikasi industri karena akurasinya yang tinggi dan fleksibilitasnya dalam menangani berbagai jenis data (Wade dan Glynn, 2020).

2.4.3 *Catboost*

Catboost adalah termasuk model yang menggunakan menggunakan algoritma *boosting* yang dirancang oleh Yandex. Sesuai dengan penamaannya yaitu *Categorical Boosting*, model ini dapat unggul dalam penanganan fitur kategori tanpa memerlukan *one-hot encoding*. CatBoost menggunakan pendekatan *depth-wise*. Dalam pendekatan ini, pohon dibangun dengan memperluas cabang sedalam mungkin sebelum beralih ke cabang lain. Ini berarti pohon tumbuh secara vertikal, dan setiap pemisahan dipilih untuk meminimalkan fungsi kerugian pada kedalaman tertentu sebelum berpindah ke kedalaman berikutnya (Ibrahim dkk, 2020).

2.4.4 *LightGBM*

Light Gradient Boosting Machine (LightGBM) adalah salah satu algoritma *machine learning* yang juga termasuk dalam keluarga algoritma *boosting*. LightGBM menggunakan pendekatan yang memperluas pohon secara *leaf-wise*, berbeda dengan *level-wise* seperti yang dilakukan oleh XGBoost. LightGBM membangun pohon secara *leaf-wise*, artinya pohon tumbuh dengan memperluas daun yang memiliki penurunan *loss* terbesar. Ini memungkinkan pohon untuk tumbuh lebih dalam dan lebih cepat dalam area tertentu yang dianggap paling penting oleh model (Ponsam dkk, 2021).

2.4.5 *Support Vector Machine*

Salah satu contoh model *Non-Tree-Based* yaitu *Support Vector Machine* (SVM). Untuk melakukan prediksi perlu dibentuk *hyperplane* optimal yang dapat memisahkan dua kelas dalam data sedemikian rupa sehingga memaksimalkan margin. Margin adalah jarak antara *hyperplane* dan titik data terdekat dari masing-masing kelas (*support vectors*)(Pisner dan Schnyer, 2020). Margin maksimal bisa diketahui menggunakan persamaan (1).

$$J(w) = \frac{1}{2}||w||^2 + C\left[\frac{1}{N}\sum_1^n \max(0,1 - y_i * (w \,.\, x_i + b)\right] \quad (1)$$

Keterangan:
J(w): Cost Function
C: Parameter Regularisasi
N: Banyak data
y: Label data
x: data

2.4.6 *Gaussian Naive Bayes*

Gaussian Naive Bayes merupakan salah satu algoritma klasifikasi *machine learning* yang memanfaatkan penghitungan probabilitas dan distribusi gaussian (persamaan (2) hingga (4)). Di dalam model ini, diasumsikan bahwa distribusi dari nilai-nilai tiap fitur yang diamati dalam setiap kelas merupakan distribusi normal. Hasil prediksi didapat dari hasil posterior terbesar dari hasil perkalian distribusi gaussian dengan prior (Reddy dkk, 2022).

- *Prior*

$$P(c) = \frac{jumlah\ data\ pada\ kelas\ c}{total\ jumlah\ data} \quad (2)$$

- Distribusi Gaussian

$$P(w|c) = \frac{1}{\sqrt{2\pi{\sigma_c}^2}} e^{\left(\frac{-(w-\mu_c)^2}{2{\sigma_c}^2}\right)} \quad (3)$$

Keterangan:
π: 3.14159265359…
${\sigma_c}^2$: varians fitur w pada kelas c
w: nilai fitur pada data uji c
μ_c: rata-rata fitur w pada kelas c

- Posterior :

$$P(c|w) = P(c) * P(w|c) \quad (4)$$

Keterangan:
P(c): Prior tiap kelas
P(w|c): Distribusi Gaussian

2.4.7 *K-Nearest Neighbors*

KNN adalah salah satu model *machine learning* yang bukan berbasis pohon keputusan. Model ini akan berusaha mencari sebanyak K tetangga terdekat dari data yang akan dilakukan prediksi. Penghitungan jarak terdekat dilakukan dengan berbagai macam metrik jarak seperti *Manhattan Distance*, *Euclidian Distance*, *Chebyshev Distance*, dan *Minkowski Distance* yang masing-masing ditunjukkan pada persamaan (5) hingga (8) (Sabry, 2023).

- ***Manhattan Distance***

$$Distance = \sum_{i=1}^{n} |x_i - y_i| \quad (5)$$

- ***Euclidian Distance***

$$Distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (6)$$

- ***Chebyshev Distance***

$$Distance = \max(\lceil x_2 - x_1 \rceil, \lceil y_2 - y_1 \rceil) \quad (7)$$

- ***Minkowski Distance***

$$Distance = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}} \quad (8)$$

2.4.8 *Optuna*

Optuna digunakan untuk mengotomasi pencarian hyperparameter guna menemukan nilai hyperparameter yang optimal dengan memanfaatkan sampler yang berbeda (Hassanali dkk, 2024). Ketujuh model ini diuji dengan tujuan untuk menemukan nilai akurasi dan hasil prediksi terbaik. Parameter yang akan diuji pada optuna dapat dilihat pada Tabel 2 hingga 7.

Tabel 2. Rentang Parameter Model *Random Forest*

Parameter	Range
n_estimators	100-350
max_depth	10-32
min_samples_split	2-20
min_samples_leaf	1-10
bootstrap	True, False

Tabel 3. Rentang Parameter Model LightGBM

Parameter	Range
n_estimators	100-350
max_depth	3-10
min_child_samples	1-20
subsample	0.5-1.0
colsample_bytree	0.5-1.0
learning_rate	0.001 - 0.1
reg_alpha	1e-5-10.0
reg_lambda	1e-5-10.0

Tabel 4. Rentang Parameter Model Catboost

Parameter	Range
n_estimators	100-350
max_depth	3-10
min_child_samples	1-20
colsample_bylevel	0.5-1.0
learning_rate	0.001-0.1
reg_lambda	1e-5-10.0

Tabel 5. Rentang Parameter Model XGBoost

Parameter	Range
n_estimators	100, 350
max_depth	3, 10
min_child_samples	1, 20
colsample_bylevel	0.5, 1.0
learning_rate	0.001, 0.1
reg_alpha	1e-5, 10.0
reg_lambda	1e-5, 10.0
min_child_weight	1,10
subsample	0.5, 1.0

Tabel 6. Rentang Parameter Tuning Model SVM

Parameter	Range
C	1e-2-1e+2, log=True
kernel	'linear', 'poly', 'rbf', 'sigmoid'
gamma	'scale', 'auto'
degree	2-5

Tabel 7. Rentang Parameter Model KNN

Parameter	Range
n_neighbors	3, 15
weights	'uniform', 'distance'
metric	'minkowski', 'euclidean', 'manhattan', 'chebyshev'

2.5 Evaluasi

Tahap evaluasi adalah proses setelah model dilatih untuk menilai kinerjanya terhadap data yang belum pernah diketahui. Tujuan utama evaluasi adalah untuk memahami seberapa baik model dapat menggeneralisasi pengetahuan dari data latih (training data) ke data uji (testing data). Langkah-langkah evaluasi meliputi pembagian data dengan *Train-Test-Split*, pelatihan model, prediksi, dan evaluasi prediksi.

Tiga metrik evaluasi yang digunakan yaitu akurasi, AUC-ROC, dan F1-score Micro. Akurasi adalah metrik yang mengukur proporsi prediksi yang benar dibandingkan dengan keseluruhan prediksi yang dibuat oleh model (Vujović, 2021). AUC-ROC mengukur kemampuan model untuk membedakan antara kelas positif dan negatif pada berbagai ambang batas (Carrington dkk, 2022). F1-score Micro cocok untuk situasi dengan ketidakseimbangan kelas. Dalam perhitungan F1-score Micro, semua *true positives* (TP), *false positive* (FP), dan *false negative* (FN) dari setiap kelas digabung terlebih dahulu kemudian precision dan recall dihitung dari total ini (Grandini dkk, 2020). *Precision* dan *recall* adalah dua metrik evalusi yang digunakan untuk mengukur kinerja model klasifikasi, terutama dalam konteks deteksi kelas positif (Miao dan Zhu, 2022). Ketiga metrik ini memberikan gambaran komprehensif tentang kinerja model dalam berbagai konteks dan kondisi data.

Metode *K-fold Validation* juga digunakan untuk evaluasi dengan membagi *dataset* menjadi beberapa bagian (*fold*) dalam rangka melakukan evaluasi model yang lebih seimbang. Metode ini dtterapkan dengan tujuan mendapatkan estimasi kinerja model yang lebih akurat dan menghindari adanya *overfitting* (Nti dkk, 2021).

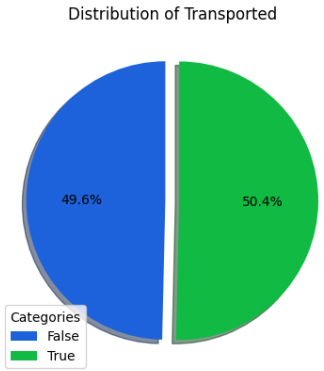
3. HASIL DAN PEMBAHASAN

3.1 Hasil Eksplorasi Data *Spaceship Titanic*

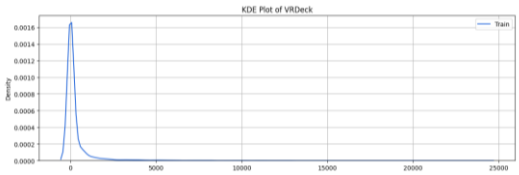
Dapat dilihat pada Tabel 8 bahwa terdapat nilai yang hilang pada 12 dari 13 fitur dalam *dataset Spaceship*. Namun, jika dihitung rata-rata persentase data yang hilang adalah sekitar 2,06%. Ini menunjukkan bahwa sebagian besar fitur memiliki data yang cukup lengkap untuk analisis lebih lanjut.

Tabel 8. Informasi Fitur *dataset Spaceship* Titanic: Tipe Data dan Nilai Hilang

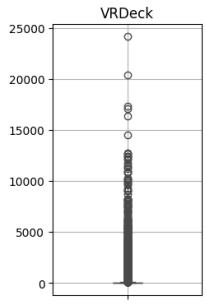
Fitur	Tipe Data	*Missing Value*
PassengerId	object	0
HomePlanet	object	201
CyroSleep	object	217
Cabin	object	199
Destination	object	182
Age	float64	179
VIP	object	203
RoomService	float64	181
FoodCourt	float64	183
ShoppingMall	float64	208
Spa	float64	183
VRDeck	float64	188
Name	object	200



Gambar 2. Distribusi Kelas *Transported* dari *Spaceship Titanic*



Gambar 3. Grafik Distribusi VRDeck – *Spaceship Titanic*



Gambar 4. Boxplot VRDeck – *Spaceship Titanic*

Berdasarkan Gambar 2 persentase kedua kelas cukup berimbang dengan hanya selisih 0,8%. Tindakan khusus seperti *oversampling* atau *undersampling* tidak diperlukan pada *dataset* ini. Namun, metode *K-fold validation* dan *Cross Validation* tetap diperlukan untuk memperoleh hasil evaluasi model yang lebih baik.

Dari hasil yang ditemukan, ternyata terjadi inflasi angka pada 5 dari 6 fitur numerik yang ada di dalam *dataset*. Sebagai salah satu contoh pada Gambar 3 dan 4, inflasi angka 0 terjadi pada fitur VRDeck. Akibatnya angka selain 0 pada fitur ini dianggap sebagai data pencilan. Oleh karena itu, tidak dilakukan penanganan apapun terhadap anomali ini.

3.2 Hasil Eksplorasi Data *Predict Failure Keep it Dry*

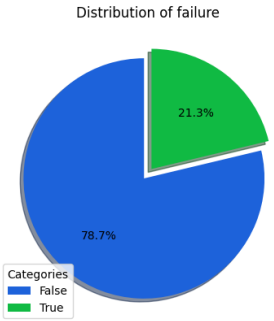
Tabel 9 menunjukkan bahwa dari 25 fitur, 16 di antaranya memiliki data yang hilang. Keseluruhan fitur yang memiliki nilai yang hilang adalah data numerik. Persentase rata-rata jumlahnya adalah sebesar 3,05%. Ini menunjukkan bahwa sebagian besar fitur memiliki data yang cukup lengkap untuk analisis lebih lanjut.

Berdasarkan Gambar 5, persentase jumlah kedua kelas tidak berimbang. Untuk itu evaluasi menggunakan *K-fold Validation* sangat direkomendasikan untuk data ini agar proporsi kelas minoritas memilki jumlah yang sama saat melakukan pelatihan model.

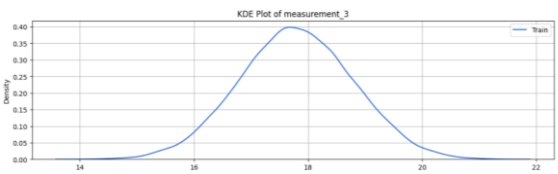
Dari visualisasi yang ditemukan pada *dataset* semua fitur numerik yang ada memiliki distribusi normal. Sebagai contoh pada Gambar 6 dan 7 pada fitur *measurement_3*.

Tabel 9. Informasi Fitur *dataset Predict Failure Keep it Dry*: Tipe Data dan Nilai Hilang

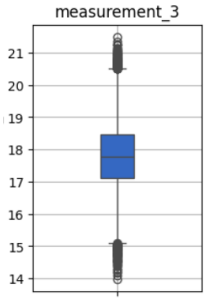
Fitur	Tipe Data	*Missing Value*
loading	float64	250
measurement_3	float64	381
measurement_4	float64	538
measurement_5	float64	676
measurement_6	float64	796
measurement_7	float64	937
measurement_8	float64	1048
measurement_9	float64	1227
measurement_10	float64	1300
measurement_11	float64	1468
measurement_12	float64	1601
measurement_13	float64	1774
measurement_14	float64	1874
measurement_15	float64	2009
measurement_16	float64	2110
measurement_17	float64	2284



Gambar 5. Distribusi Kelas *outcome* dari *Predict Failure Keep it Dry*



Gambar 6. Distribusi *measurement_3 – Predict Failure Keep it Dry*



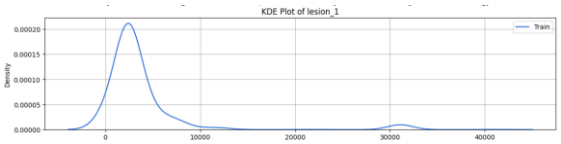
Gambar 7. Boxplot measurement_3 – *Keep it Dry*

3.3 Hasil Eksplorasi Data *Horse Health*

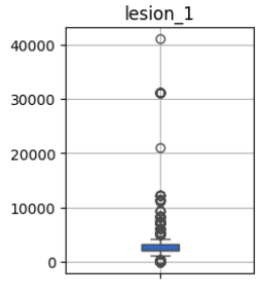
Tabel 10 menunjukkan bahwa dari 29 fitur, 12 di antaranya terdapat data yang hilang. Keseluruhan fiturnya adalah data kategori. Persentase rata-rata jumlahnya adalah sebesar 2,14%. Ini menunjukkan bahwa sebagian besar fitur memiliki data yang cukup lengkap untuk analisis lebih lanjut.

Tabel 10. Informasi Fitur *dataset Horse Health*: Tipe Data dan Nilai Hilang

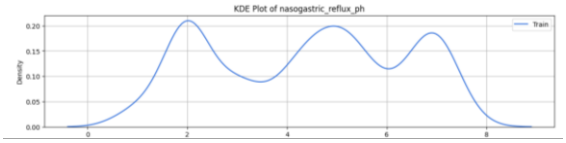
Fitur	Tipe Data	*Missing Value*
temp_of_extremities	object	39
peripheral_pulse	object	60
mucous_membrane	object	21
capillary_refill_time	object	6
pain	object	44
peristalsis	object	20
abdominal_distention	object	23
nasogastric_tube	object	80
nasogastric_reflux	object	21
rectal_exam_feces	object	190
abdomen	object	213
abdomo_appearance	object	48



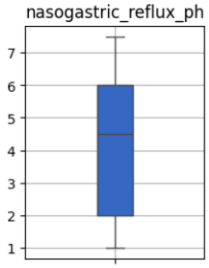
Gambar 8. Distribusi *lesion_1 – Horse Health*



Gambar 9. Boxplot *lesion_1 – Keep it Dry*



Gambar 10. Distribusi *nasogastric_reflux_ph – Horse Health*



Gambar 11. Boxplot *nasogastric_reflux_ph – Horse Health*

Tidak semua data numerik pada *dataset Horse Health* memiliki distribusi yang baik. Terdapat dua fitur yang memiliki kasus inflasi angka, yaitu *lesion_1* dan *nasogastric_reflux_ph* (lihat pada Gambar 8 hingga Gambar 11). Selain itu, fitur *nasogastric_reflux_ph* juga memiliki kasus multimodal seperti pada Gambar 10 dan 11.

3.4 Hasil Prediksi *Spaceship Titanic*

Pada Tabel 11 dapat dilihat bahwa hasil metrik evaluasi tertinggi untuk akurasi terdapat pada model LightGBM dengan nilai 0,814836. Sedangkan untuk metrik AUC-ROC tertinggi ada pada model Catboost dengan nilai 0,896464.

Berbeda dengan menggunakan *train test split* Tabel 12 dapat dilihat bahwa hasil metrik evaluasi tertinggi untuk akurasi ada pada model XGBoost dengan nilai 0,814836. Sedangkan untuk metrik AUC-ROC dan F1-score Micro tertinggi ada pada model catboost dengan nilai 0,896464.

3.5 Hasil Prediksi *Predict Failure Keep it Dry*

Hasil prediksi menggunakan *dataset Predict Failure Keep it Dry* dapat dilihat pada Tabel 13. Berdasarkan hasil evaluasi, model CatBoost menunjukkan performa tertinggi dalam hal akurasi dan F1-score dengan nilai masing-masing sebesar 0,801468. Sementara itu, model XGBoost memiliki nilai tertinggi pada metrik AUC-ROC dengan skor 0,592468. Hal ini menunjukkan bahwa CatBoost unggul dalam metrik yang mengukur ketepatan dan keseimbangan klasifikasi, sedangkan XGBoost lebih baik dalam kemampuan memisahkan kelas pada kurva ROC.

Di sisi lain ternyata model Naive Bayes memiliki hasil metrik AUC-ROC tertinggi kedua setelah model XGBoost. Temuan ini mengindikasikan bahwa untuk *dataset* ini, CatBoost, XGBoost, dan Gaussian NB dapat menjadi pilihan model yang sangat efektif, masing-masing dengan keunggulan dalam metrik evaluasi yang berbeda.

Tabel 11. Hasil Prediksi *Spaceship Titanic* menggunakan *Train Test Split*

Model	Akurasi	AUC-ROC	F1-score Micro
LightGBM	**0,814836**	0,896455	**0,814836**
Random Forest	0,799885	0,886578	0,799885
XGBoost	0,805635	0,896464	0,805635
Catboost	0,810236	**0,898912**	0,810236
SVM	0,799310	0,878231	0,799310
GaussianNB	0,706728	0,844707	0,706728
KNN	0,763657	0,836574	0,763657

Tabel 12. Hasil Prediksi *Spaceship Titanic* menggunakan *K-fold Validation*

Model	Akurasi	AUC-ROC	F1-score Micro
LightGBM	0,805589	0,899263	0,805589
Random Forest	0,802829	0,893468	0,802829
XGBoost	**0,810766**	0,899591	0,810766
Catboost	0,813643	**0,901113**	**0,813643**
SVM	0,801103	0,883934	0,801103
GaussianNB	0,705165	0,850850	0,705165
KNN	0,762682	0,839928	0,762682

Tabel 13. Hasil Prediksi *Predict Failure Keep it Dry* menggunakan *Train Test Split*

Model	Akurasi	AUC-ROC	F1-score
LGBM	0,801091	0,586127	0,801091
Random Forest	0,801280	0,558410	0,801280
XGB	0,801091	**0,592468**	0,801091
CatBoost	**0,801468**	0,551737	**0,801468**
SVC	0,801091	0,492700	0,801091
NB	0,796575	0,590330	0,796575
KNN	0,799586	0,531647	0,799586

Tabel 14. Hasil Prediksi *Predict Failure Keep it Dry* menggunakan *K-fold Validation*

Model	Akurasi	AUC-ROC	F1-score Micro
LGBM	**0,787392**	0,582056	**0,787392**
Random Forest	0,787354	0,554787	0,787354
XGB	**0,787392**	**0,586028**	**0,787392**
CatBoost	0,786601	0,552677	0,786601
SVC	**0,787392**	0,503450	**0,787392**
NB	0,782198	0,577575	0,782198
KNN	0,785698	0,524725	0,785698

Hasil Prediksi *Predict Failure Keep it Dry* menggunakan *K-fold Validation* dapat dilihat pada Tabel 14. Dengan menggunakan *K-fold*, hasil evaluasi model LGBM, XGBoost, dan SVC menunjukkan performa tertinggi dalam hal akurasi dan F1-Score dengan nilai yang sama yaitu 0,787392. Namun, XGBoost memiliki nilai tertinggi pada metrik AUC-ROC dengan skor 0,586028. Hal ini menunjukkan bahwa LGBM, XGBoost, dan SVC sama-sama unggul dalam hal ketepatan dan keseimbangan klasifikasi, sementara XGBoost

memiliki kemampuan terbaik dalam memisahkan kelas pada kurva ROC. Temuan ini mengindikasikan bahwa untuk tugas klasifikasi tertentu, XGBoost bisa menjadi pilihan utama karena performa unggul dalam dua metrik evaluasi, terutama dalam kemampuan memisahkan kelas.

3.6 Hasil Prediksi *Horse Health*

Tabel 15 menunjukkan hasil prediksi *Horse Health* menggunakan *Train Test Split*. Dengan membagi data menjadi data *test* dan data *train* dengan rasio 8:2, CatBoost memiliki performa tertinggi untuk ketiga metrik yang diuji. Sehingga dapat dianggap sebagai model terbaik berdasarkan data yang diberikan.

Hasil Prediksi *Horse Health* menggunakan *K-fold Validation* ditunjukkan pada Tabel 16. Berdasarkan hasil evaluasi, model Catboost menunjukkan performa tertinggi dalam hal akurasi, AUC-ROC, dan F1-Score dengan nilai masing-masing sebesar 0,813643, 0,901113, dan 0,813643. Hal ini menunjukkan bahwa Catboost unggul dalam ketepatan klasifikasi, kemampuan memisahkan kelas pada kurva ROC, dan keseimbangan klasifikasi. Temuan ini mengindikasikan bahwa untuk tugas klasifikasi tertentu, Catboost merupakan pilihan model yang sangat efektif dan dapat memberikan hasil yang optimal di berbagai metrik evaluasi

Tabel 15. Hasil Prediksi *Horse Health* menggunakan *Train Test Split*

Model	Akurasi	AUC-ROC	F1-score
LGBM	0,704453	0,842326	0,704453
Random Forest	0,692308	0,844277	0,692308
XGB	0,700405	0,846823	0,700405
CatBoost	**0,720648**	**0,852928**	**0,720648**
SVC	0,700405	0,841259	0,700405
NB	0,408907	0,750547	0,408907
KNN	0,672065	0,812051	0,672065

Tabel 16. Hasil Prediksi *Horse Health* menggunakan *K-fold Validation*

Model	Akurasi	AUC-ROC	F1-score Micro
LightGBM	0,805589	0,899263	0,805589
Random Forest	0,802829	0,893468	0,802829
XGBoost	0,810766	0,899591	0,810766
Catboost	**0,813643**	**0,901113**	**0,813643**
SVM	0,801103	0,883934	0,801103
GaussianNB	0,705165	0,850850	0,705165
KNN	0,762682	0,839928	0,762682

4. KESIMPULAN

Berdasarkan hasil evaluasi, beberapa temuan kunci dapat disimpulkan. Model CatBoost secara konsisten menunjukkan performa tertinggi di berbagai metrik, termasuk akurasi, AUC-ROC, dan F1-score, yang menunjukkan keandalannya dan efektivitasnya dalam tugas klasifikasi. Sebagai contoh, CatBoost mencapai akurasi dan F1-Score tertinggi dalam skenario validasi *K-fold* dengan nilai 0,813643 serta AUC-ROC tertinggi sebesar 0,901113. Selain itu, XGBoost juga menunjukkan performa yang sangat baik, terutama dalam metrik AUC-ROC, yang menyoroti kemampuannya untuk secara efektif memisahkan kelas dalam *dataset*. XGBoost secara konsisten membuktikan keandalannya dengan mendapatkan skor tinggi dalam berbagai pengaturan evaluasi.

Hasil tidak terduga dapat disimpulkan dari model SVM dan GaussianNB yang beberapa kali mengungguli beberapa model berbasis pohon keputusan. Hal ini menunjukkan bahwa dalam kondisi data tertentu, model *non-tree-based* seperti SVM dan Naive Bayes dapat memberikan performa yang kompetitif, bahkan mengalahkan model berbasis pohon keputusan. Sebagai contoh, pada beberapa kasus, GaussianNB menunjukkan metrik AUC-ROC yang cukup baik, dan SVM mencapai performa akurasi dan F1-Score yang sebanding dengan model *tree-based*.

Secara keseluruhan, penelitian ini menunjukkan bahwa untuk tugas klasifikasi tertentu, model CatBoost merupakan pilihan yang sangat efektif dan dapat memberikan hasil yang optimal di berbagai metrik evaluasi. Namun, model *non-tree-based* seperti SVM dan GaussianNB juga tidak boleh diabaikan karena mereka dapat memberikan performa yang kompetitif tergantung pada karakteristik data yang digunakan.

DAFTAR PUSTAKA

ALPAYADIN, E., 2020. Introduction to Machine Learning, Fourth Edition. Cambridge: MIT Press.

AMPOMAH, E.K., QIN, Z. dan NYAME, G., 2020. Evaluation of tree-based ensemble machine learning models in predicting stock price direction of movement. Information, 11(6), hal. 332.

CARRINGTON, A.M. dkk., 2022. Deep ROC analysis and AUC as balanced average accuracy, for improved classifier selection, audit and explanation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(1), hal. 329–341.

FATYANOSA, T.N. dan BACHTIAR, F.A., 2017. Classification method comparison on Indonesian social media sentiment analysis. 2017 International Conference on Sustainable Information Engineering and Technology, hal. 310–315. Tersedia di: https://doi.org/10.1109/SIET.2017.8304154

GENUER, R. dan POGGI, J.-M., 2020. Random Forests. In: Random Forests with R. Use R!.

Cham: Springer International Publishing, hal. 33–55. Tersedia di: https://doi.org/10.1007/978-3-030-56485-8_3.

GRANDINI, M., BANGLI, E. dan VISANI, G., 2020. Metrics for Multi-Class Classification: an Overview. arXiv. Tersedia di: http://arxiv.org/abs/2008.05756 [Diakses 22 Jul. 2024].

HASSANALI, M. dkk., 2024. Software development effort estimation using boosting algorithms and automatic tuning of hyperparameters with Optuna. Journal of Software: Evolution and Process, hal. e2665. Tersedia di: https://doi.org/10.1002/smr.2665.

IBRAHIM, A.A. dkk., 2020. Comparison of the CatBoost classifier with other machine learning methods. International Journal of Advanced Computer Science and Applications, 11(11). Tersedia di: https://pdfs.semanticscholar.org/948c/ae886bd76ab222f3be431ab0a71e6aa03286.pdf [Diakses 22 Jul. 2024].

JAYATILAKE, S.M.D.A.C. dan GANEGODA, G.U., 2021. Involvement of Machine Learning Tools in Healthcare Decision Making. Journal of Healthcare Engineering. Disunting oleh M. MARTORELLI, 2021, hal. 1–20. Tersedia di: https://doi.org/10.1155/2021/6679512.

JIANG, M. dkk., 2020. An improved Stacking framework for stock index prediction by leveraging tree-based ensemble models and deep learning algorithms. Physica A: Statistical Mechanics and its Applications, 541, hal. 122272.

MIAO, J. dan ZHU, W., 2022. Precision–recall curve (PRC) classification trees. Evolutionary Intelligence, 15(3), hal. 1545–1569. Tersedia di: https://doi.org/10.1007/s12065-021-00565-2.

NTI, I.K., NYARKO-BOATENG, O. dan ANING, J., 2021. Performance of machine learning algorithms with different K values in K-fold CrossValidation. International Journal of Information Technology and Computer Science, 13(6), hal. 61–71.

PISNER, D.A. dan SCHNYER, D.M., 2020. Support vector machine', in Machine learning. Elsevier, hal. 101–121. Tersedia di: https://www.sciencedirect.com/science/article/pii/B9780128157398000067 [Diakses 22 Jul. 2024].

PONSAM, J.G. dkk., 2021. Credit Risk Analysis using LightGBM and a comparative study of popular algorithms. 2021 4th International Conference on Computing and Communications Technologies (ICCCT), hal. 634–641. Tersedia di: https://ieeexplore.ieee.org/abstract/document/9711896/ [Diakses 22 Jul. 2024].

REDDY, E.M.K. dkk., 2022. Introduction to Naive Bayes and a review on its subtypes with applications. Bayesian reasoning and gaussian processes for machine learning applications, hal. 1–14.SABRY, F., 2023. K Nearest Neighbor Algorithm: Fundamentals and Applications. One Billion Knowledgeable.

VUJOVIĆ, Ž., 2021. Classification model evaluation metrics. International Journal of Advanced Computer Science and Applications, 12(6), hal. 599–606.

WADE, C. dan GLYNN, K., 2020. Hands-On Gradient Boosting with XGBoost and scikit-learn: Perform accessible machine learning and extreme gradient boosting with Python. Packt Publishing Ltd. Tersedia di: https://books.google.com/books?hl=id&lr=&id=2tcDEAAAQBAJ&oi=fnd&pg=PP1&dq=what+is+xgboost&ots=s5sLInmmmO&sig=UJAptEfWVJZ3QRGVkfXUq7xUov8 [Diakses 22 Jul. 2024].

WHIG, P., GUPTA, K. dan JIWANI, N., 2022. Real-Time Detection of Cardiac Arrest Using Deep Learning. AI-Enabled Multiple-Criteria Decision-Making Approaches for Healthcare Management. IGI Global, hal. 1–25. Tersedia di: https://www.igi-global.com/chapter/real-time-detection-of-cardiac-arrest-using-deep-learning/312326 [Diakses 22 Jul. 2024].

Halaman ini sengaja dikosongkan