Sebuah Studi Perbandingan Memprediksi Penyakit Jantung dengan Algoritma Machine Learning

1. Introduction

Salah satu penyebab kematian paling umum di dunia adalah penyakit jantung. Untuk meningkatkan peluang perawatan yang efektif, penting untuk mendeteksi penyakit ini sejak dini. Dengan menggunakan machine learning, proyek ini berfokus pada prediksi penyakit jantung dan membantu tenaga medis mengidentifikasi pasien yang memiliki risiko tinggi terkena serangan jantung. Teknik prediktif seperti ini dapat mengurangi biaya perawatan kesehatan dan menyelamatkan nyawa.

Menyediakan alat bantu yang cepat dan akurat bagi dokter adalah alasan utama untuk menyelesaikan masalah ini. Dengan menggunakan model machine learning, prediksi berdasarkan berbagai faktor kesehatan pasien dapat dilakukan secara real-time. Variabel seperti usia, jenis kelamin, tekanan darah, kadar kolesterol, dan variabel lainnya dapat dimasukkan ke dalam model.

Input dan Output

Input dari masalah ini adalah karakteristik klinis pasien, seperti usia, jenis kelamin, tekanan darah, kadar kolesterol, detak jantung maksimal, dan faktor lainnya.

Outputnya adalah prediksi risiko penyakit jantung seseorang. Kita dapat memprediksi kemungkinan pasien terkena penyakit jantung dengan menggunakan berbagai algoritma machine learning, seperti Logistic Regression, K-Nearest Neighbors (KNN), Random Forest, Adaboost, dan Support Vector Classifier (SVC).

Proyek ini berbeda dari yang lain karena menggunakan pendekatan beragam model serta mengombinasikan beberapa teknik untuk mengatasi masalah ketidakseimbangan data. Penggunaan Hyperparameter Tuning seperti GridSearchCV dan RandomizedSearchCV juga dilakukan untuk mengatasi tantangan tersebut.

Tujuan akhir dari proyek ini adalah menemukan model yang memiliki performa prediksi terbaik dan paling akurat pada data baru. Model ini dapat digunakan dalam situasi nyata di mana dokter membantu dalam pengambilan keputusan.

2. Related Work

Beberapa penelitian sebelumnya telah dilakukan menggunakan metode yang serupa untuk mendukung pengembangan model prediksi penyakit jantung ini. Berikut ini adalah beberapa karya yang relevan:

1. **Machine Learning-Based Predictive Models for Detection of Cardiovascular Diseases:** Paper ini membahas model machine learning yang digunakan untuk prediksi penyakit jantung. Model-model ini termasuk K-Nearest Neighbors, Support Vector Machine, Logistic Regression, dan XGBoost. Penelitian ini membahas masalah dengan dataset yang tidak seimbang, yang sebanding dengan masalah yang mungkin saya temui selama proyek ini dikerjakan. Dengan XGBoost, hasil terbaik mencapai akurasi 98.50%. (Link)

2. **Heart Disease Prediction Using Machine Learning Methods:** Jurnal yang ditulis oleh Birmingham City University menggunakan berbagai algoritma machine learning untuk memprediksi penyakit jantung. Algoritma-algoritma ini termasuk Random Forest dan Gradient Boosting. Analisis menyeluruh terhadap dataset yang seimbang dan tidak seimbang diberikan dalam kertas yang dapat diakses secara gratis. (Link)

3. **A Review of Machine Learning Algorithms for Heart Disease Prediction:** Penelitian ini menjelaskan kekuatan dan kelemahan berbagai algoritma dan teknik optimasi untuk deteksi dini penyakit jantung, seperti Logistic Regression, Decision Trees, dan kumpulan teknik seperti AdaBoost. (Link)

4. **Predictive Analytics for Heart Disease Using Machine Learning:** Fokus artikel ini adalah evaluasi model prediksi jantung yang berbeda menggunakan teknik seperti SVM dan Random Forest. Teknik-teknik ini relevan dengan proyek ini karena membahas bagaimana standarisasi data dapat mempengaruhi kinerja model. (Link)

5. **Heart Disease Prediction Using Hybrid Machine Learning Models:** Dalam artikel ini membahas penggunaan model hybrid untuk meningkatkan kinerja model prediksi dengan menggunakan metode seperti KNN dan SVM. Jika model individual tidak memberikan hasil yang optimal, model hybrid dapat menjadi pilihan yang baik. (Link)

Kategori Pendekatan

- **Algoritma Klasifikasi Tradisional (Logistic Regression, SVM, KNN):** Penelitian tentang kategori ini menekankan kecepatan dan interpretasi yang baik, namun seringkali tidak ideal untuk dataset yang besar dan kompleks seperti prediksi penyakit jantung.

- **Ensemble Learning (Random Forest, XGBoost, dan AdaBoost):** Kategori ini berfokus pada penggunaan berbagai model untuk meningkatkan akurasi, dan cocok untuk dataset yang kompleks dan tidak seimbang. Meskipun hasilnya menunjukkan kinerja yang lebih baik dalam beberapa situasi, mereka mungkin memerlukan sumber daya komputasi yang lebih besar.

Kelebihan dan Kekurangan

- Metode kelompok seperti Random Forest dan XGBoost memiliki keuntungan dalam menangani dataset besar dan memberikan akurasi yang tinggi, tetapi mereka memiliki kekurangan, yaitu waktu komputasi yang lebih lama dan model yang lebih sulit untuk diinterpretasikan.

- Kelebihan algoritma tradisional seperti Logistic Regression dan KNN adalah mereka mudah dipahami dan melakukan prediksi dengan cepat. Namun, kekurangannya, terutama untuk dataset yang tidak seimbang, adalah tingkat akurasi yang lebih rendah.

Persamaan dan Perbedaan

- Persamaan: Baik metode yang saya gunakan maupun yang dibahas dalam paper ini memanfaatkan dataset yang sebanding (UCI Kesehatan Jantung), dan menghadapi masalah data yang tidak seimbang.

- Perbedaan: Ketika saya menggunakan model individu seperti Logistic Regression, KNN, dan SVC, saya menggunakan model ensemble atau hybrid, sedangkan paper yang saya referensikan fokus pada optimasi.

Opini:

Logistic Regression, Random Forest, KNN, AdaBoost, dan SVC adalah model yang saya gunakan sesuai dengan metode prediksi penyakit jantung yang umum. Karena ketahanannya terhadap overfitting dan peningkatan generalisasi pada dataset medis yang terstruktur, sebagian besar makalah menekankan pentingnya metode ensemble seperti Random Forest dan AdaBoost.

Namun, seperti yang ditunjukkan dalam beberapa makalah, kinerja prediktif yang lebih baik dapat dicapai dengan menggabungkan metode voting ensemble bersama dengan metode peningkatan lainnya. Selain itu, seperti yang ditunjukkan oleh penelitian ini, berkonsentrasi pada penyesuaian hyperparameter menggunakan cross-validation kemungkinan akan meningkatkan kinerja dan keandalan model. Tuning dan pengujian yang cermat sangat penting untuk memaksimalkan akurasi model pada proyek prediksi penyakit jantung karena hasil yang beragam dari berbagai dataset.

3. Dataset & Features

Dataset penyakit jantung dari UCI Machine Learning Repository terdiri dari 303 sampel dengan 13 fitur input, termasuk usia, jenis kelamin, tekanan darah, kolesterol, dan hasil pemeriksaan klinis lainnya. Dataset ini digunakan untuk proyek ini.

Hingga saat ini, satu-satunya database yang telah digunakan oleh peneliti machine learning adalah database Cleveland. Eksperimen yang dilakukan dengan database Cleveland berfokus pada upaya untuk membedakan antara keberadaan penyakit jantung (nilai 1, 2, 3, 4) dan tidak adanya penyakit (nilai 0). Kolom "target" memiliki nilai integer dari 0 hingga 4 yang menunjukkan keberadaan penyakit jantung pada pasien.

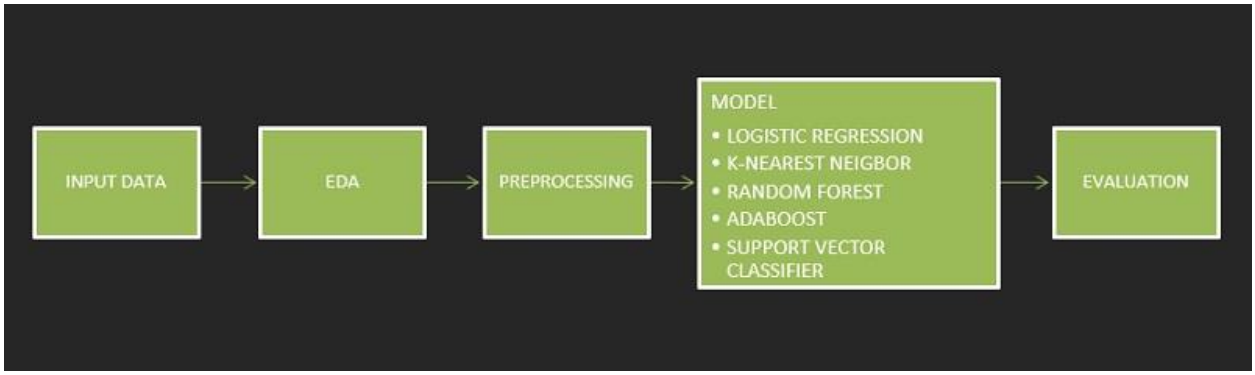
Dataset dibagi menjadi dua bagian: 80% untuk training dan 20% untuk testing. Proses preprocessing mencakup imputasi nilai missing, standarisasi data dengan `RobustScaler`, serta One-Hot Encoding untuk variabel kategorikal. Selain itu, saya melakukan teknik oversampling untuk mengatasi masalah ketidakseimbangan kelas target.

Dataset Atribut

Atribut	Type	Deskripsi
age	Integer	Usia pasien
sex	Categorical	Jenis kelamin pasien *(0=wanita, 1=pria)*
cp	Categorical	Tipe nyeri dada *(1=angina tipikal, 2=angina atipikal, 3=nyeri non-angina, 4=tanpa gejala)*
trestbps	Integer	Tekanan darah istirahat (mm Hg)
chol	Integer	Kolesterol serum (mg/dl)
fbs	Categorical	Gula darah puasa >120 mg/dl (1=benar, 0=salah)
restecg	Categorical	Hasil elektrokardiografi istirahat *(0=normal, 1=memiliki kelainan gelombang ST-T (inversi gelombang T dan/atau elevasi atau depresi ST > 0,05 mV), 2=menunjukkan hipertrofi ventrikel kiri yang mungkin atau pasti berdasarkan kriteria Estes)*
thalach	Integer	Denyut jantung max yang dicapai
exang	Categorical	Angina yang disebabkan oleh olahraga *(1=ya, 0=tidak)*
oldpeak	Integer	Depresi ST yang diinduksi oleh olahraga relatif terhadap istirahat
slope	Categorical	Kemiringan segmen ST puncak latihan (0-2)
ca	Integer	Jumlah pembuluh besar yang diwarnai oleh fluoroskopi (0-3)
thal	Categorical	Thalassemia *(1 = normal; 2 = cacat tetap; 3 = cacat reversibel)*
target	Integer	Diagnosa penyakit jantung *(0=tidak ada, 1-4=ada)*

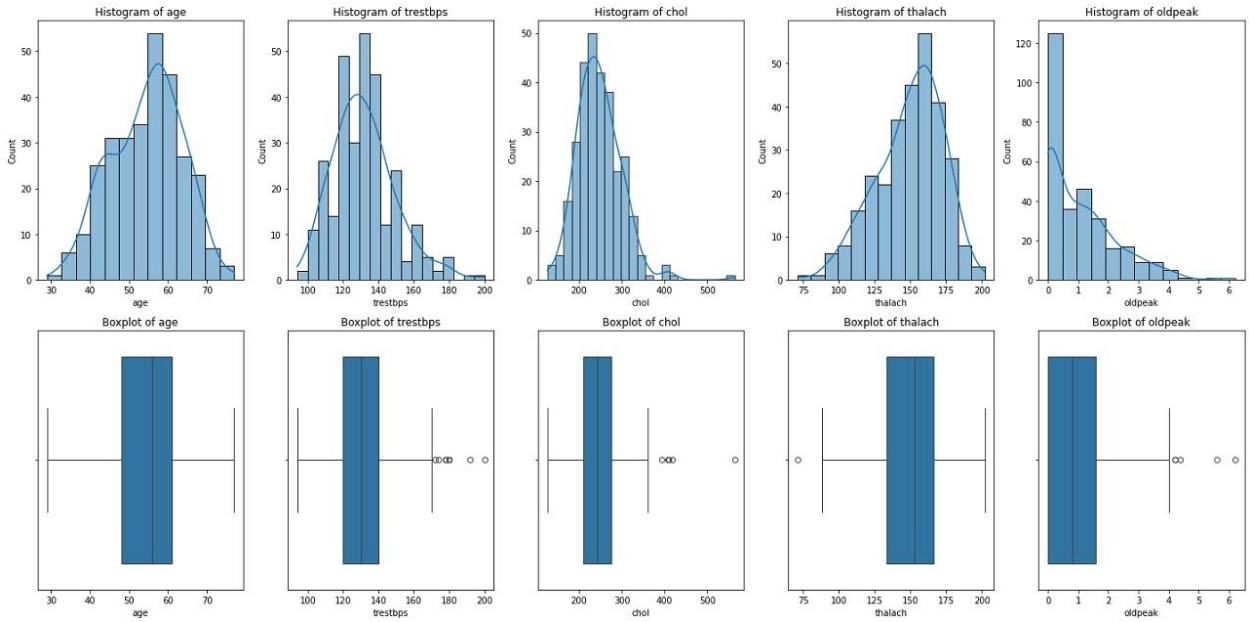
4. Methodology

Dalam bagian ini, saya memperkenalkan metode untuk membangun model prediksi menggunakan beberapa algoritma machine learning. Gambar alur di bawah menunjukkan prosedur metodologi yang digunakan untuk menentukan seberapa besar kemungkinan kita dapat mengandalkan survei awal untuk mendiagnosis penyakit jantung sebelum melakukan tes lebih lanjut.



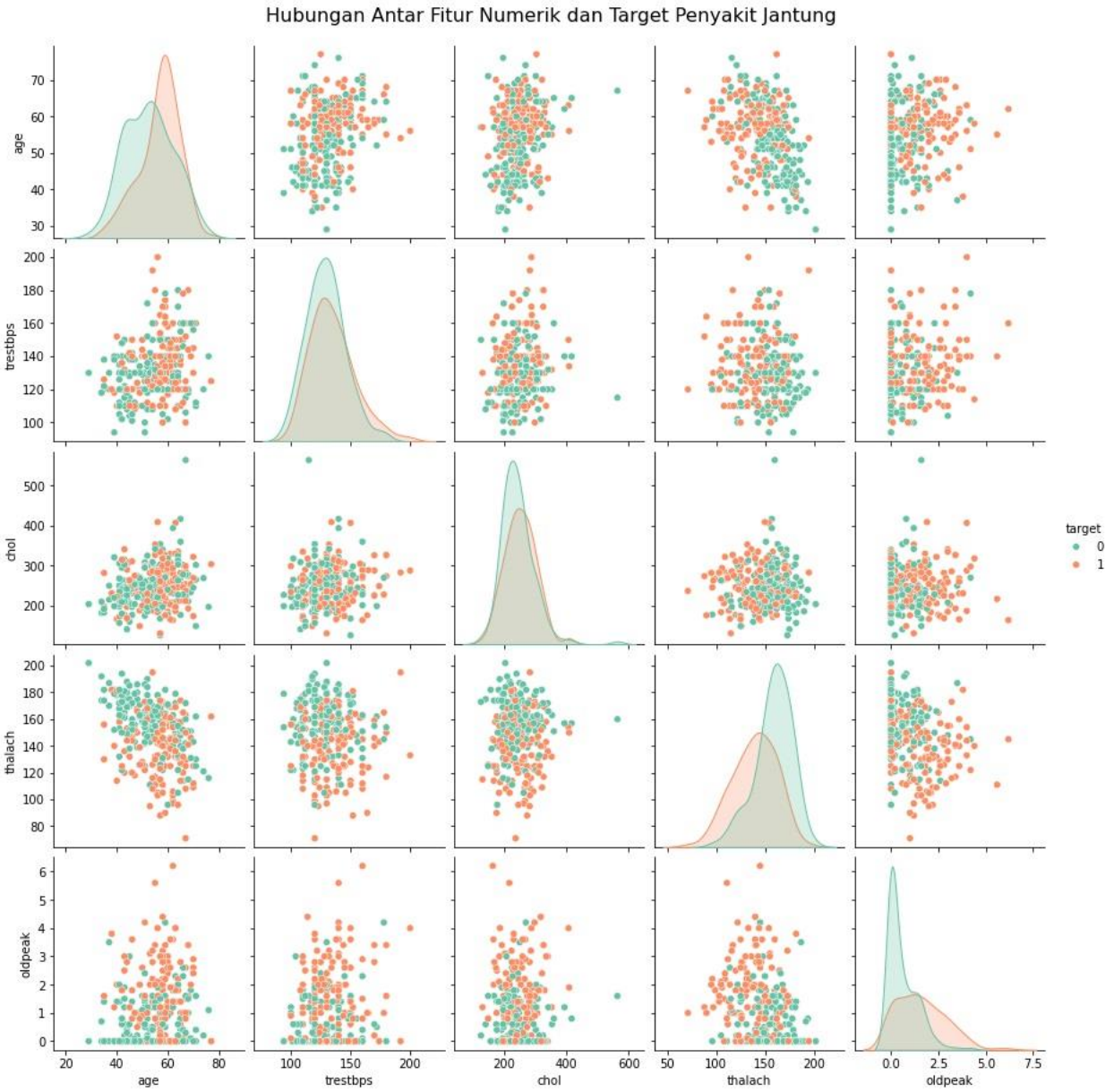
4.1 Exploratory Data Analysis

Sebelum melakukan analisis statistik, permodelan prediktif lebih lanjut diperlukan. Ini karena kita perlu memahami karakteristik, struktur, dan elemen penting dari dataset ini, yang membantu kita memahami konteksnya, menemukan pertanyaan yang relevan, dan memilih metode analisis yang tepat.



Insight:

- Distribusi: Sebagian besar variabel numerik, seperti usia, tekanan darah, kolesterol, dan denyut jantung maksimum, cenderung memiliki distribusi yang tidak simetris dengan ekor kanan yang lebih panjang, yang menunjukkan bahwa ada beberapa outlier atau nilai ekstrim pada variabel tersebut.
- Outlier: Tekanan darah dan kolesterol adalah beberapa variabel yang menunjukkan variasi yang signifikan. Jika tidak ditangani dengan benar, outlier ini dapat berdampak pada analisis statistik dan model prediksi.
- Variabilitas: Variabel seperti usia dan kolesterol mewakili perbedaan yang signifikan di antara pasien.

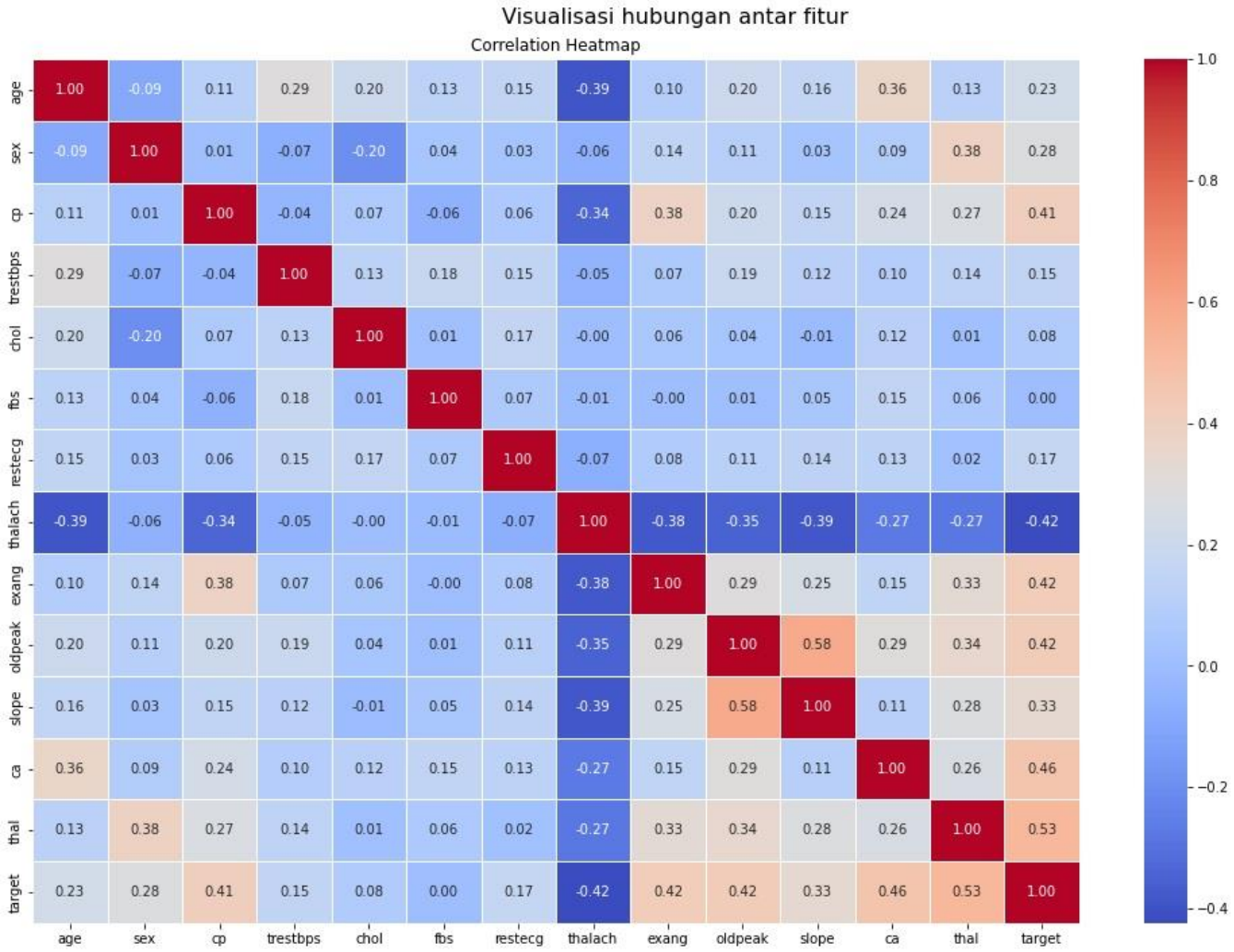


Insight:

- Distribusi Variabel:
    - Sebagian besar variabel numerik, seperti usia, tekanan darah, kolesterol, denyut jantung maksimum, dan depresi ST, cenderung memiliki distribusi yang tidak simetris, dengan ekor kanan yang lebih panjang, yang menunjukkan adanya beberapa outlier atau nilai ekstrim pada variabel-variabel tersebut.
    - Variabel target (penyakit jantung), terdistribusi cukup seimbang, dengan jumlah pasien dengan dan tanpa penyakit jantung relatif sama.
- Hubungan Antar Variabel:
    - Variabel-variabel ini memiliki beberapa hubungan yang menarik. Misalnya, kadar kolesterol dan usia meningkatkan risiko penyakit jantung.

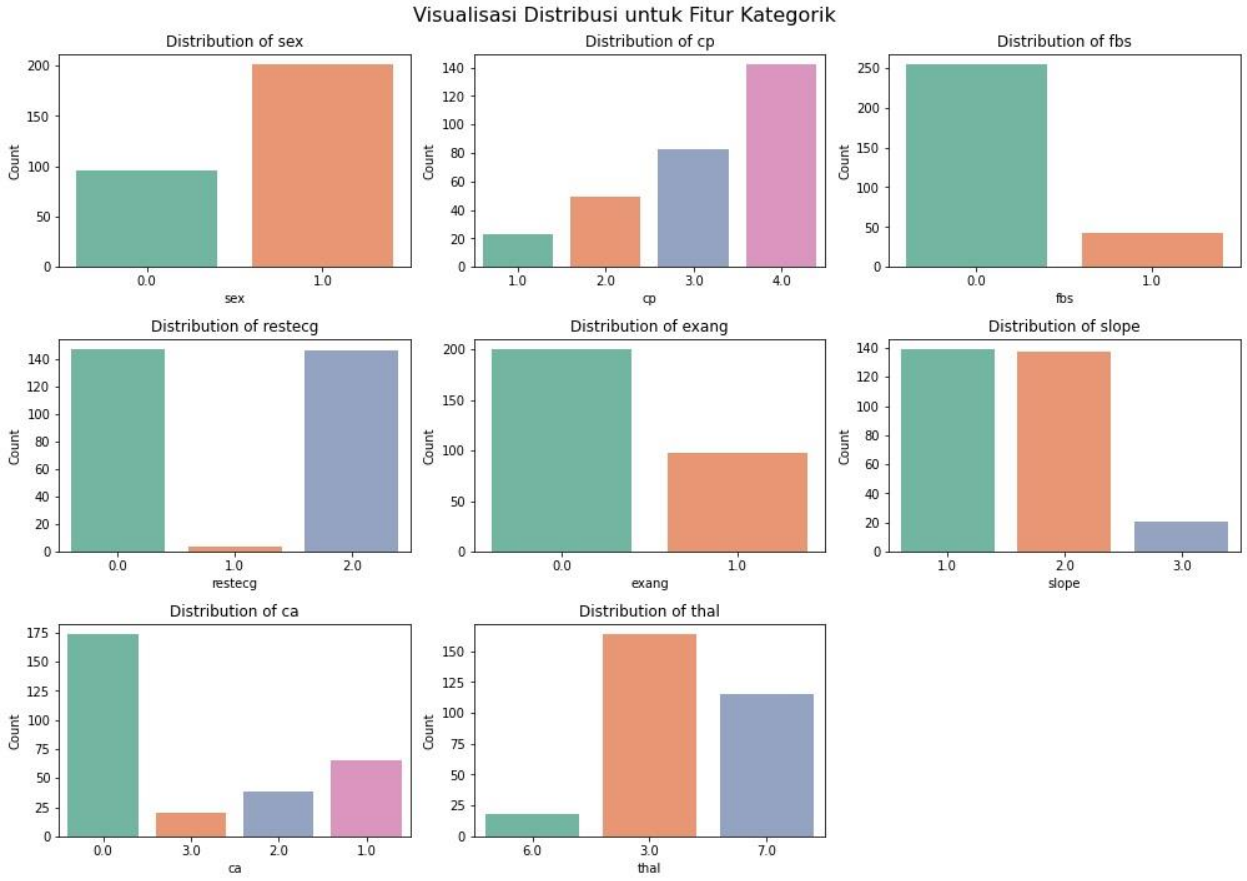
- Selain itu, telah ditemukan korelasi signifikan antara tekanan darah, denyut jantung maksimum, dan risiko penyakit jantung.
- Outlier: Beberapa plot menunjukkan outlier, terutama pada variabel kolesterol dan depresi ST. Jika tidak ditangani dengan baik, outlier ini dapat mempengaruhi analisis dan model prediksi.

- Interpretasi
  - Usia: Kemungkinan terkena penyakit jantung meningkat seiring bertambahnya usia seseorang.
  - Tekanan Darah: Ada korelasi antara tingginya tekanan darah dan risiko penyakit jantung.
  - Kolesterol: Salah satu faktor risiko utama penyakit jantung adalah kadar kolesterol tinggi.
  - Denyut Jantung Maksimum: Denyut jantung maksimum yang rendah dapat menunjukkan masalah jantung atau risiko penyakit jantung.
  - Depresi ST: Nilai depresi ST yang lebih tinggi dapat menunjukkan masalah jantung atau risiko penyakit jantung.



Insight:

- Korelasi Positif Kuat:
    - Usia dan penyakit jantung: Semakin tua usia seseorang, semakin tinggi risiko terkena penyakit jantung.
    - Kolesterol dan penyakit jantung: Ada korelasi antara kadar kolesterol tinggi dan risiko penyakit jantung.
    - Depresi ST dan penyakit jantung: Nilai depresi ST yang lebih tinggi dapat menjadi indikasi masalah jantung atau risiko penyakit jantung.
    - Meskipun korelasi yang ditunjukkan di atas tidak sekuat yang ditunjukkan sebelumnya, beberapa variabel lain juga menunjukkan korelasi positif dengan penyakit jantung.
- Korelasi Negatif:
    - Thalach (denyut jantung maksimum) dan penyakit jantung: Penyakit jantung seringkali dikaitkan dengan denyut jantung maksimum yang rendah.



Insight:

- sex: Sebagian besar data adalah laki-laki (nilai 1).
- cp: Terdapat beberapa tipe nyeri dada (cp), dengan tipe 3 (nyeri non-angina) yang paling sering muncul.
- fbs: Sebagian besar pasien memiliki gula darah puasa yang normal (nilai 0).

- restecg: Hasil elektrokardiografi istirahat paling banyak menunjukkan kondisi normal (nilai 0).
- exang: Sebagian besar pasien tidak mengalami angina yang disebabkan oleh olahraga (nilai 0).
- slope: Sebagian besar pasien memiliki kemiringan segmen ST puncak latihan yang bernilai 1.
- ca: Sebagian besar pasien memiliki jumlah pembuluh besar yang diwarnai oleh fluoroskopi bernilai 0.
- thal: Sebagian besar pasien memiliki nilai thal bernilai 3.

4.2 Model yang Digunakan:

Saya menggunakan beberapa model machine learning untuk prediksi, di antaranya:

1. Logistic Regression: digunakan untuk klasifikasi biner. Dengan menggunakan fungsi sigmoid, model ini memprediksi kemungkinan terjadinya suatu peristiwa. Batasan nilainya berkisar antara 0 dan 1.
    - Algoritma: Metode Maximum Likelihood Estimation (MLE) digunakan untuk mengoptimalkan model. Tujuannya adalah untuk menemukan parameter yang memaksimalkan kemungkinan mengamati data yang diberikan.
    - Cara Kerja: Logistic regression menggunakan turunan gradien untuk memperbarui parameter secara iteratif hingga ditemukan titik di mana loss function (negatif dari log-likelihood) mencapai minimum.
    - Notasi Matematis:
        - X: Fitur input
        - β: Koefisien regresi yang dipelajari
        - σ(z): Fungsi sigmoid

$$P(y = 1 \mid X) = \sigma(X\beta) = \frac{1}{1 + e^{-X\beta}}$$

2. K-Nearest Neighbors (KNN): adalah algoritma non-parametrik untuk klasifikasi yang bekerja berdasarkan jarak antara sampel.
    - Algoritma:
        - Hitung jarak antara data uji dengan semua data latih
        - Pilih k tetangga terdekat
        - Klasifikasi dilakukan berdasarkan mayoritas label dari tetangga terdekat
    - Cara Kerja: Algoritma menghitung jarak dari tiap data ke sampel uji, kemudian memprediksi berdasarkan suara terbanyak dari k tetangga terdekat
    - Notasi Matematis:
        - *d(x,x ')*: Fungsi jarak, umumnya Euclidean distance:

$$d(x, x') = \sqrt{\sum_{i=1}^{n} (x_i - x_i')^2}$$

3. Random Forest: adalah algoritma ensemble yang menggabungkan banyak decision trees untuk menghasilkan prediksi yang lebih akurat dan stabil.
    - Algoritma:
        - Buat banyak decision trees menggunakan bootstrap sampling.
        - Setiap pohon hanya menggunakan subset acak dari fitur.
        - Gabungkan prediksi semua pohon menggunakan majority vote (untuk klasifikasi) atau rata-rata (untuk regresi).
    - Cara Kerja: Random Forest memitigasi overfitting yang sering terjadi pada single decision tree dengan mengombinasikan banyak pohon yang dilatih dari subset acak data dan fitur
    - Notasi Matematis:
        - *pk:* Proporsi kelas k pada node tersebut
        - Setiap decision tree membagi data berdasarkan informasi gain atau Gini impurity. Gini impurity untuk node *j* dihitung sebagai:

$$G(j) = 1 - \sum_{k=1}^{K} p_k^2$$

4. AdaBoost: adalah algoritma ensemble yang meningkatkan kinerja model lemah secara bertahap dengan memberikan bobot lebih besar pada data yang sulit diklasifikasikan.
    - Algoritma:
        - Latih model dasar pada data.
        - Hitung kesalahan dan perbarui bobot data.
        - Data yang salah diklasifikasikan diberi bobot lebih tinggi pada iterasi selanjutnya.
    - Cara kerja AdaBoost adalah untuk melatih model lemah secara iteratif dengan fokus pada sampel yang salah diklasifikasikan. Ini memungkinkan model untuk berkonsentrasi pada kesalahan dari model sebelumnya.
    - Notasi Matematis:
        - αt: Bobot model dasar pada iterasi t, yang dihitung berdasarkan error dari model tersebut
        - Misalkan ht(x) adalah model dasar pada iterasi t, maka total prediksi adalah:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

5. Support Vector Classifier (SVC): bertujuan untuk menemukan hyperplane yang memisahkan kelas-kelas dalam ruang fitur dengan margin maksimum.
    - Algoritma:
        - Cari w dan b yang memaksimalkan margin.
        - Jika data tidak bisa dipisahkan secara linear, gunakan kernel trick untuk memetakan data ke dimensi yang lebih tinggi.
    - Cara kerja: SVC memisahkan data dengan margin maksimum, dan untuk data yang tidak terpisahkan secara linear, kernel (seperti RBF atau polynomial) digunakan untuk memetakan ke ruang fitur yang lebih tinggi agar dapat dipisahkan.
    - Notasi matematis:
        - Hyperplane didefinisikan sebagai wTx+b=0, dengan margin dihitung sebagai:

$$\text{Margin} = \frac{||w||}{2}$$

5. Experiments / Result / Discussion

Eksperiment:

- Penjelasan Hyperparameter yang Dipilih
  Dalam rekap evaluasi model, GridSearchCV dan RandomizedSearchCV digunakan untuk mengubah hyperparameter. Kedua metode ini menggunakan cross-validation (CV) untuk memilih kombinasi hyperparameter yang optimal.
    - GridSearchCV secara eksplisit mencoba setiap kombinasi hyperparameter yang disediakan. Pada proyek ini ini terlihat dari hasil RandomForestClassifier dan SVC yang memiliki beberapa parameter yang diujikan, seperti max_depth, n_estimators, dan min_samples_split.
    - RandomizedSearchCV mencoba menggabungkan hyperparameter secara acak dari distribusi yang ditentukan; teknik ini lebih cepat daripada GridSearchCV, terutama dalam situasi di mana ruang pencarian hyperparameter sangat besar.

  Kedua metode ini biasanya menggunakan cross-validation (CV) dengan kolom k, yang membagi data menjadi beberapa subset. Dalam kasus CV 5-fold, data dibagi menjadi 5 bagian, 4 untuk pelatihan dan 1 untuk validasi, dan ini diulang 5 kali, sehingga setiap subset hanya digunakan untuk validasi sekali. Karena memberikan keseimbangan antara waktu pelatihan dan kinerja

model, cross-validation yang umum adalah CV 5-fold atau 10-fold. Namun saya menggunakan GridSearchCV default biasanya 5-fold.
Hyperparameter yang dipilih:

- Logistic Regression
  - Best Param Grid: {'C': 0.01, 'max_iter': 100, 'penalty': 'l2', 'solver': 'liblinear'}
  - Best Param Random: {'solver': 'liblinear', 'penalty': 'l2', 'max_iter': 300, 'C': 0.01}
  - Pada model Logistic Regression, parameter C mengontrol regularisasi. Nilai kecil (seperti 0.01) memberikan regularisasi yang lebih besar untuk mengurangi overfitting. max_iter adalah jumlah iterasi maksimum untuk konvergensi model, dan penalty='l2' mengindikasikan regularisasi ridge.

- KNeighborsClassifier:
  - Best Param Grid: {'metric': 'euclidean', 'n_neighbors': 9, 'weights': 'distance'}
  - Best Param Random: {'weights': 'distance', 'n_neighbors': 9, 'metric': 'euclidean'}
  - n_neighbors menentukan jumlah tetangga terdekat yang akan dipertimbangkan untuk klasifikasi. weights='distance' membuat tetangga yang lebih dekat memiliki bobot lebih besar.

- Random Forest Classifier:
  - Best Param Grid: {'max_depth': 30, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 300}
  - Best Param Random: {'n_estimators': 300, 'min_samples_split': 10, 'min_samples_leaf': 1, 'max_depth': None}
  - Hyperparameter seperti max_depth (kedalaman pohon) membatasi kompleksitas pohon untuk mencegah overfitting. min_samples_leaf dan min_samples_split menentukan minimal sampel di node atau saat memecah node. n_estimators adalah jumlah pohon.

- Support Vector Classifier (SVC):
  - Best Param Grid: {'C': 0.1, 'gamma': 'auto', 'kernel': 'sigmoid'}
  - Best Param Random: {'kernel': 'sigmoid', 'gamma': 'auto', 'C': 0.1}
  - Parameter C mengontrol margin SVC, di mana nilai kecil mendorong margin yang lebih lebar (lebih banyak salah klasifikasi). gamma menentukan seberapa jauh pengaruh satu sampel meluas (semakin besar, semakin sempit). Kernel sigmoid digunakan untuk memperkenalkan non-linearitas.

- Metrik Utama yang Digunakan
  Untuk evaluasi model, beberapa metrik utama digunakan:
  - Accuracy (Akurasi): Metrik yang paling umum untuk klasifikasi, menghitung persentase prediksi yang benar dari total prediksi. Formula:

$$\text{Accuracy} = \frac{\text{True Positives (TP)} + \text{True Negatives (TN)}}{\text{Total Samples}}$$

- Precision: Metrik yang mengukur ketepatan prediksi positif, yakni proporsi dari prediksi positif yang benar. Formula:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{Faslse Positif (FP)}}$$

- Recall (Sensitivity): Mengukur seberapa baik model mendeteksi kelas positif yang sebenarnya. Formula:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{Faslse Negatif (FP)}}$$

- F1 Score: Kombinasi dari precision dan recall, memberikan keseimbangan ketika ada ketidakseimbangan antara positive dan negative class. Formula:

$$\text{F1 Score} = 2 \text{ x } \frac{\text{Precision x Recall}}{\text{Precision} + \text{Recall}}$$

- AUC (Area Under the Curve): Metrik ini mengukur seberapa baik model dapat membedakan antara kelas. Semakin dekat nilai AUC ke 1, semakin baik model dalam memprediksi kelas yang benar.

- Confusion Matrix: Matriks yang menggambarkan performa klasifikasi dari model, mengidentifikasi TP, FP, TN, dan FN.

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

  Ini memberikan informasi mendalam tentang kesalahan model, seperti apakah model banyak memprediksi false positif atau false negatif.

- Evaluasi Model Berdasarkan Metrik

  Logistic Regression dan Random Forest mencapai hasil yang sangat baik pada rekapan model, masing-masing dengan akurasi sekitar 0.83 - 0.85 pada set tes. Namun, Random Forest tampaknya overfit karena akurasi pelatihannya yang sangat tinggi (1.00), Sebaliknya, hasil test AdaBoostClassifier dan SVC tidak stabil.

  Untuk mengevaluasi keseimbangan antara precision dan recall, Confusion Matrix dan ROC-AUC curves juga diperlukan. Dalam klasifikasi medis seperti prediksi penyakit jantung, ini sangat penting karena FN (false negatives) yang tinggi dapat menyebabkan diagnosis yang terlewatkan, yang dapat fatal dalam hal medis.

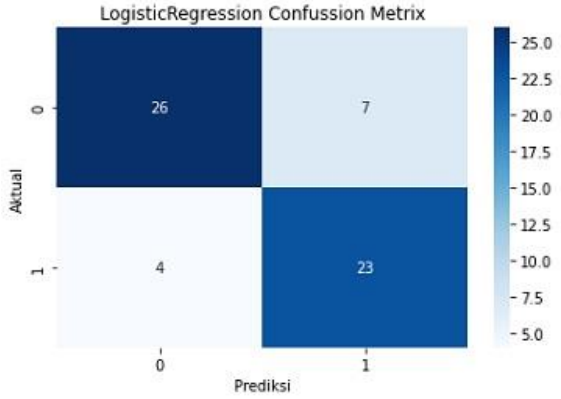
- Overfitting dan Solusi

  Overfitting terjadi ketika model bekerja dengan sangat baik pada data pelatihan tetapi buruk pada data tes (data yang belum pernah dilihat sebelumnya). Ini terlihat jelas pada Random Forest, di mana akurasi pelatihan 100%, menunjukkan bahwa model telah mempelajari data yang terlalu spesifik, termasuk bias.

  Untuk mengatasi overfitting:
  1. Untuk mengurangi kompleksitas model, ubah parameter RandomForest seperti max_depth dan n_estimators.
  2. Regularization: Untuk membatasi kompleksitas model, gunakan regularization seperti Logistic Regression.
  3. Data Peningkatan: Untuk membuat model lebih umum, gunakan metode seperti cross-validation yang lebih intensif (misalnya, CV stratified atau 10-fold) atau lebih banyak data.

Result

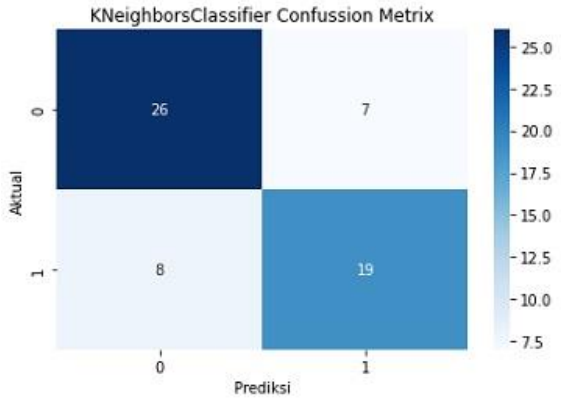
- Logistic Regression Confusion Metrix & Classification Report



LogisticRegression Classification Report

	precision	recall	f1-score	support
0	0.87	0.79	0.83	33
1	0.77	0.85	0.81	27
accuracy			0.82	60
macro avg	0.82	0.82	0.82	60
weighted avg	0.82	0.82	0.82	60

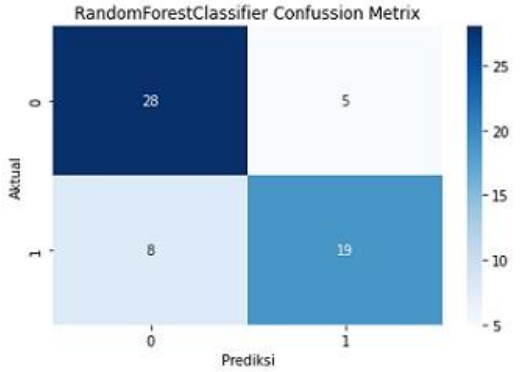
- KneighborsClassifier Confusion Metrix & Classification Report



KNeighborsClassifier Classification Report

	precision	recall	f1-score	support
0	0.76	0.79	0.78	33
1	0.73	0.70	0.72	27
accuracy			0.75	60
macro avg	0.75	0.75	0.75	60
weighted avg	0.75	0.75	0.75	60

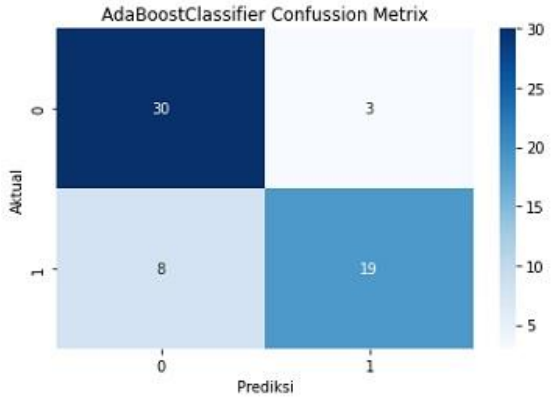
- RandomForestClassifier Confusion Metrix & Classification Report



RandomForestClassifier Classification Report

	precision	recall	f1-score	support
0	0.78	0.85	0.81	33
1	0.79	0.70	0.75	27
accuracy			0.78	60
macro avg	0.78	0.78	0.78	60
weighted avg	0.78	0.78	0.78	60

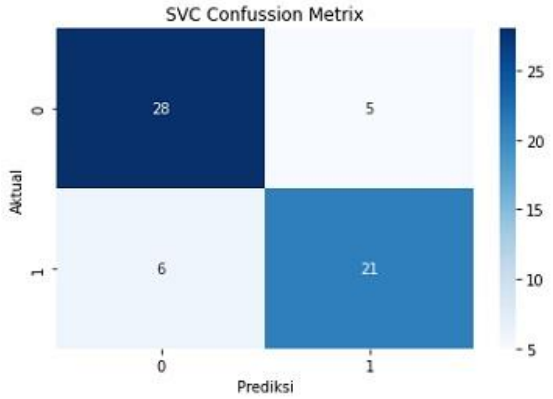
- AdaBoostClassifier Confusion Metrix & Classification Report



AdaBoostClassifier Classification Report

	precision	recall	f1-score	support
0	0.79	0.91	0.85	33
1	0.86	0.70	0.78	27
accuracy			0.82	60
macro avg	0.83	0.81	0.81	60
weighted avg	0.82	0.82	0.81	60

- SVC Confusion Metrix & Classification Report



SVC Classification Report

	precision	recall	f1-score	support
0	0.82	0.85	0.84	33
1	0.81	0.78	0.79	27
accuracy			0.82	60
macro avg	0.82	0.81	0.81	60
weighted avg	0.82	0.82	0.82	60

Rekap hasil akurasi permodelan

	Model	Accuracy (Train)	Accuracy (Test)	Best Param Grid	Best Param Random
0	LogisticRegression	0.814184	0.833333	{'C': 0.01, 'max_iter': 100, 'penalty': 'l2', 'solver': 'liblinear'}	None
1	RandomForestClassifier	0.802039	0.833333	{'max_depth': 30, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 300}	None
2	LogisticRegression	0.814184	0.833333	None	{'solver': 'liblinear', 'penalty': 'l2', 'max_iter': 300, 'C': 0.01}
3	RandomForestClassifier	0.797872	0.833333	None	{'n_estimators': 300, 'min_samples_split': 10, 'min_samples_leaf': 1, 'max_depth': None}
4	LogisticRegression	0.839662	0.816667	None	None
5	RandomForestClassifier	1.000000	0.816667	None	None
6	AdaBoostClassifier	0.907173	0.816667	None	None
7	SVC	0.877637	0.816667	None	None
8	KNeighborsClassifier	0.801596	0.816667	{'metric': 'euclidean', 'n_neighbors': 9, 'weights': 'distance'}	None
9	AdaBoostClassifier	0.810372	0.816667	{'learning_rate': 0.1, 'n_estimators': 50}	None
10	SVC	0.801950	0.816667	{'C': 0.1, 'gamma': 'auto', 'kernel': 'sigmoid'}	None
11	KNeighborsClassifier	0.801596	0.816667	None	{'weights': 'distance', 'n_neighbors': 9, 'metric': 'euclidean'}
12	AdaBoostClassifier	0.810372	0.816667	None	{'n_estimators': 50, 'learning_rate': 0.1}
13	SVC	0.801950	0.816667	None	{'kernel': 'sigmoid', 'gamma': 'auto', 'C': 0.1}
14	KNeighborsClassifier	0.827004	0.750000	None	None

Kesimpulan

- Logistic Regression adalah model yang baik untuk baseline karena performa yang konsisten di data training dan test.
- Random Forest dapat memiliki hasil yang lebih baik jika overfitting dikurangi melalui tuning lebih lanjut.
- Model seperti AdaBoost dan SVC memberikan hasil yang cukup stabil dan bisa menjadi kandidat yang baik untuk lebih dieksplorasi.