



MACHINE LEARNING-HEART DISEASE

ROFIK

TABLE OF CONTENTS

01

Definisi Problem Statement

02

Mengapa Machine Learning?

03

Pemilihan Model yang Relevan

04

Eksekusi Kode Model

05

Hasil Fitur Importance

06

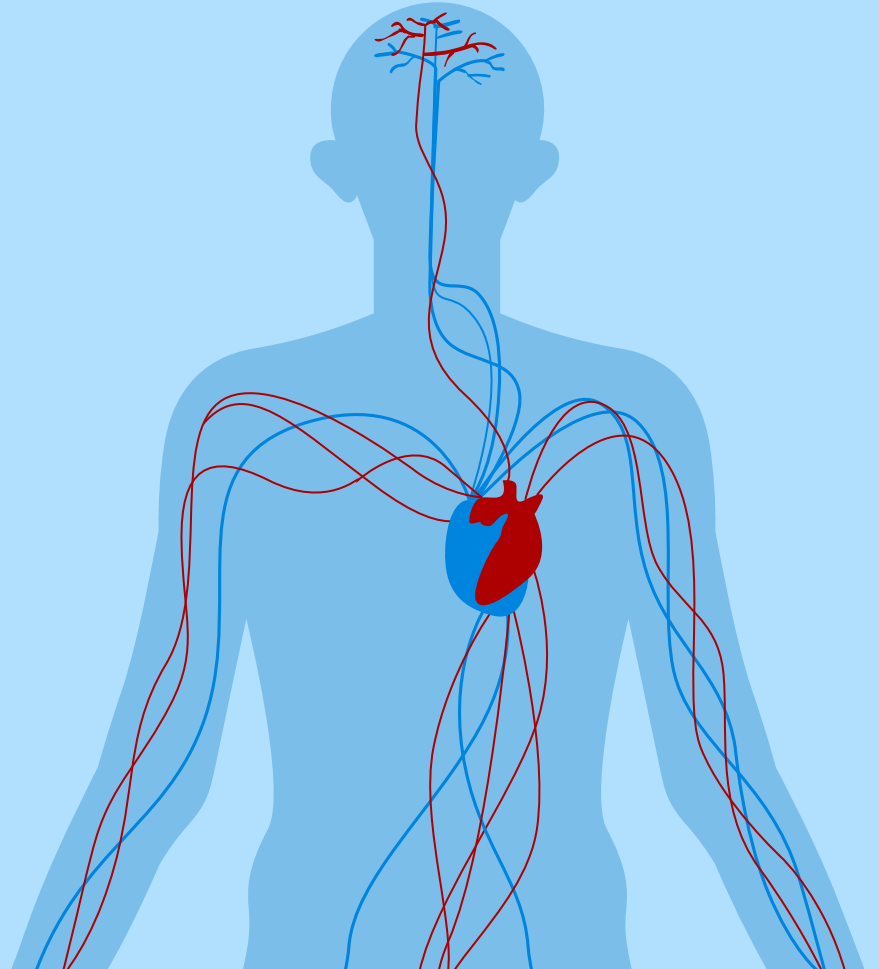
Visualisasi Fitur Importances

INTRODUCTION

Penyakit jantung, sering disebut sebagai pembunuh nomor satu di seluruh dunia, telah menjadi epidemi global yang mendalam. Pentingnya deteksi penyakit jantung menjadi semakin nyata seiring dengan meningkatnya angka kematian dan dampak negatif yang ditimbulkannya pada kualitas hidup individu. Ini adalah masalah kesehatan yang memengaruhi jutaan orang di seluruh dunia, tidak mengenal usia, jenis kelamin, atau latar belakang.

01

Definisi Problem Statement

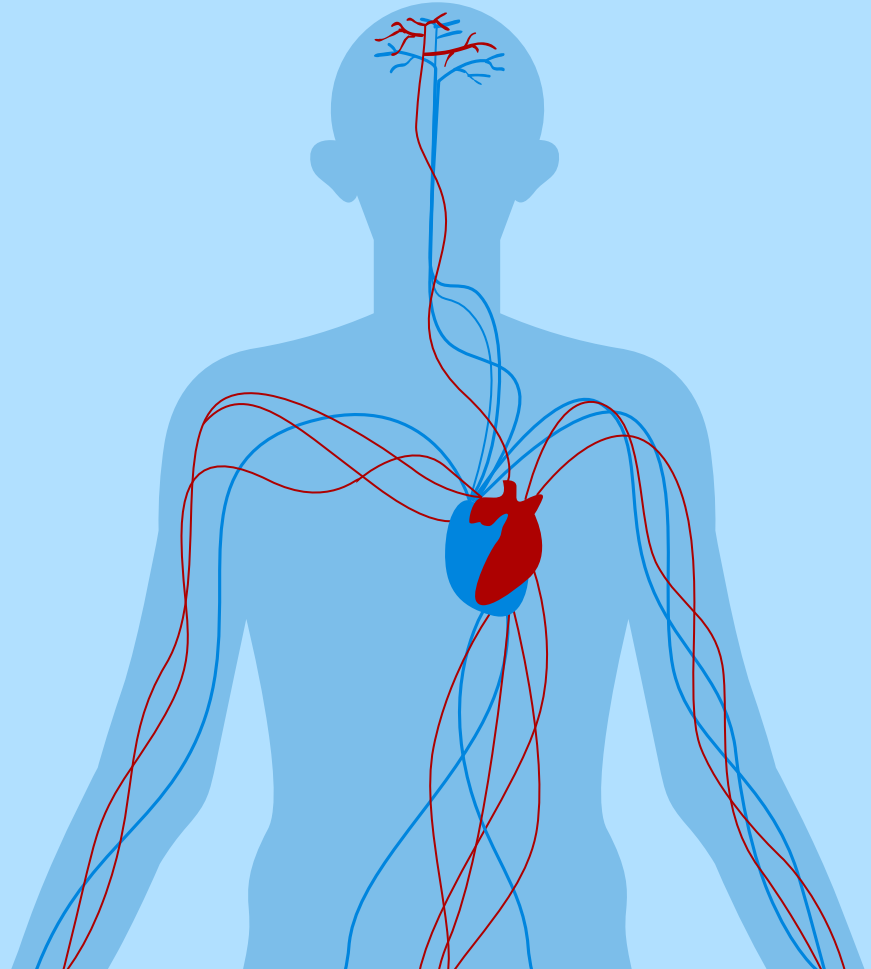


Case-Heart Disease

Tujuan dari proyek ini adalah untuk membangun model Machine Learning yang dapat memprediksi risiko penyakit jantung berdasarkan atribut medis pasien seperti usia, jenis kelamin, tekanan darah, kadar kolesterol, dan lainnya. Prediksi ini akan membantu dokter dan peneliti untuk mengidentifikasi faktor risiko yang signifikan dan mengambil langkah-langkah pencegahan yang sesuai.

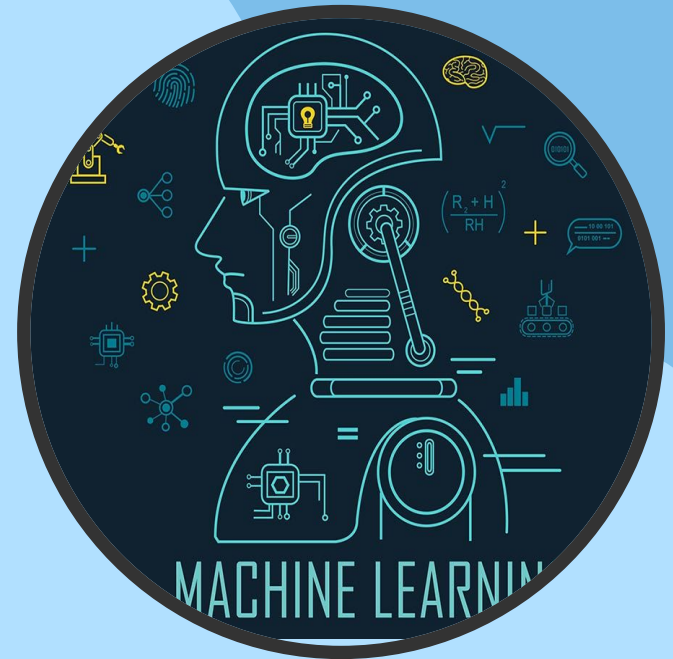
02

Mengapa Machine Learning?

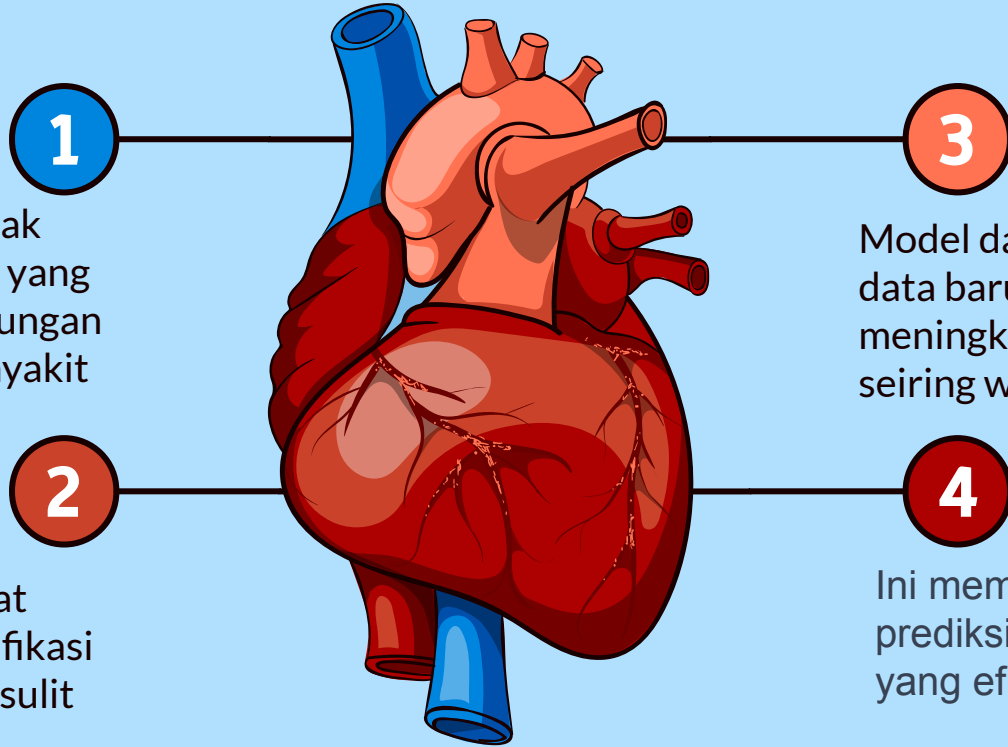


Machine Learning

Machine Learning adalah cabang dari kecerdasan buatan yang memungkinkan komputer untuk belajar dari data dan pengalaman sebelumnya untuk membuat keputusan atau melakukan tugas tanpa pemrograman eksplisit. Ini memungkinkan komputer 'mengerti' pola, mengidentifikasi tren, dan membuat prediksi dengan berdasarkan pada data yang ada. Machine Learning memiliki aplikasi luas, termasuk dalam pengenalan wajah, prediksi pasar saham, diagnosis medis, dan banyak lagi



Machine Learning dapat menjadi pendekatan yang tepat untuk masalah ini karena:



Dataset memiliki banyak atribut yang kompleks yang mungkin memiliki hubungan non-linear dengan penyakit jantung.

Machine Learning dapat membantu mengidentifikasi pola dan korelasi yang sulit dilihat oleh manusia.

Model dapat terus belajar dari data baru untuk meningkatkan akurasi seiring waktu.

Ini memungkinkan otomatisasi prediksi risiko penyakit jantung yang efisien.

Pemilihan Model Algoritma



Random Forest

Random Forest adalah salah satu algoritma yang sering memberikan hasil yang baik dalam masalah klasifikasi penyakit jantung. Ini adalah kombinasi dari banyak pohon keputusan yang menggabungkan prediksi mereka untuk hasil yang lebih kuat.



Gradient Boosting

Algoritma seperti Gradient Boosting (misalnya, XGBoost, LightGBM, atau CatBoost) adalah pendekatan lain yang kuat untuk klasifikasi penyakit jantung. Mereka bekerja dengan cara memperbaiki model secara berurutan, fokus pada sampel yang salah diklasifikasikan sebelumnya.



Jaringan Saraf Tiruan (Neural Networks):

Jaringan saraf tiruan, terutama jaringan saraf dalam (deep neural networks), dapat mengatasi masalah klasifikasi yang kompleks. Mereka sering digunakan untuk masalah medis seperti diagnosis penyakit jantung.

Pemilihan Model Algoritma



Regresi Logistik

Regresi logistik adalah pendekatan klasik yang sering digunakan dalam analisis medis. Ini memberikan interpretasi yang mudah dimengerti tentang dampak atribut pada peluang terjadinya penyakit. Apalagi algoritma ini biasa digunakan untuk kelas kategorikal yang sesuai dengan case ini



K-Nearest Neighbors (KNN)

KNN adalah metode berbasis instansi yang dapat efektif digunakan dalam klasifikasi penyakit jantung, terutama ketika terdapat pola jelas dalam data. Karena cara kerjanya yang mencoba mengklasifikasikan berdasarkan ketetanggaan.



Pohon Keputusan

Pohon keputusan dapat memberikan pemahaman yang baik tentang atribut mana yang paling berpengaruh dalam diagnosis penyakit jantung. Mereka juga dapat digunakan dalam ensemble seperti Random Forest.

Pemilihan Model Algoritma



SVM

SVM sangat cocok untuk tugas klasifikasi penyakit jantung karena dapat memberikan kejelasan dalam memisahkan pasien yang memiliki penyakit jantung dan yang tidak. Dengan menemukan hyperplane optimal yang memiliki margin terbesar, SVM memberikan prediksi yang akurat dalam diagnosis penyakit jantung. Kemampuannya dalam menangani data yang tumpang tindih dan fleksibilitas dalam pemilihan kernel membuat SVM menjadi pilihan yang kuat dalam analisis medis seperti ini.

1

Hapus data duplikat

2

Penyeimbangan data
dengan SMOTE

3

Standarisasi data dengan
Standard scaler

**Dimodelkan setelah
melalui tahap
preprocessing**

Eksekusi Model Pertama

1

Hapus data duplikat

2

Penyeimbangan data
dengan SMOTE

3

Standarisasi data dengan
Standard scaler

4

Penghapusan Outlier

**Dimodelkan setelah
melalui tahap
preprocessing**

Eksekusi Model Kedua

1

Hapus data duplikat

2

Penyeimbangan
data dengan
SMOTE

3

Standarisasi data
dengan Standard
scaler

4

Penghapusan Outlier

5

Seleksi Fitur dengan
Random Forest

**Dimodelkan setelah
melalui tahap
preprocessing**

Eksekusi Model Ketiga

Kenapa Menyeimbangkan data dengan

SMOTE?

SMOTE digunakan untuk mengatasi ketidakseimbangan kelas dalam masalah klasifikasi. Dengan menambah sampel sintetis ke kelas minoritas, SMOTE membantu meningkatkan kinerja model, mengurangi overfitting, menghilangkan bias dalam evaluasi, dan mengoptimalkan hasil klasifikasi.

Kenapa menggunakan

Standardscaler untuk Standarisasi data?

Standard Scaler digunakan untuk standarisasi data dalam machine learning. Tujuannya adalah untuk mengubah distribusi nilai-nilai atribut sehingga memiliki rata-rata 0 dan deviasi standar 1. Dengan demikian, semua atribut berada dalam skala yang serupa, sehingga memudahkan model untuk memahami dan memproses data. Standard Scaler membantu menghilangkan perbedaan skala antar atribut, yang dapat memengaruhi kinerja algoritma yang sensitif terhadap skala, seperti regresi logistik dan analisis diskriminan. Selain itu, standarisasi juga membantu mengatasi masalah outlier dan mempermudah konvergensi pada algoritma berbasis gradien. Dengan kata lain, Standard Scaler adalah langkah penting dalam pra-pemrosesan data yang dapat meningkatkan kualitas dan hasil dari model machine learning.

Kenapa menggunakan

Seleksi Fitur dengan Random Forest?

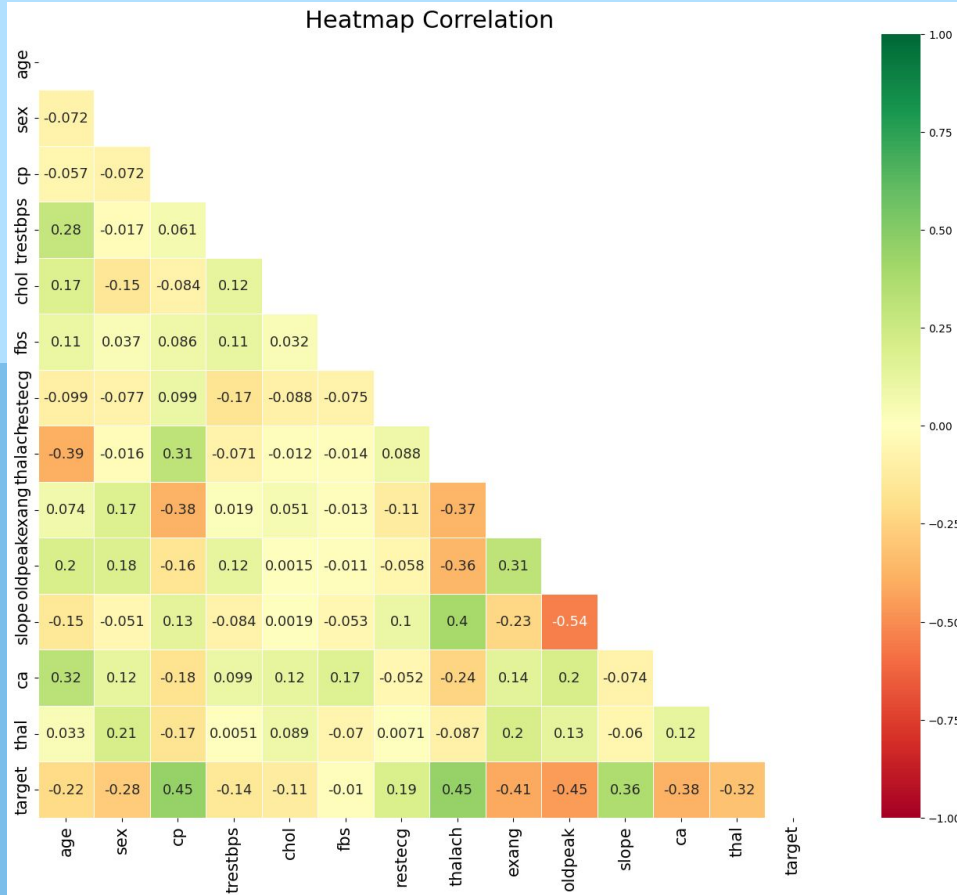
Seleksi fitur menggunakan Random Forest memberikan keunggulan dalam mengidentifikasi atribut paling penting dalam prediksi penyakit jantung. Algoritma Random Forest memilih fitur yang memberikan kontribusi terbesar dalam membedakan pasien yang memiliki penyakit jantung dengan yang tidak. Dengan menggunakan hanya fitur-fitur yang paling penting, kita dapat mempercepat pelatihan model, mengurangi overfitting, dan meningkatkan akurasi prediksi. Ini memungkinkan kita untuk fokus pada atribut yang paling relevan, menghasilkan model yang lebih efisien, dan memperkuat kemampuan kita dalam mengklasifikasikan penyakit jantung.

Kenapa menggunakan

Cross-validation?

Cross-validation digunakan untuk mengukur sejauh mana model klasifikasi dapat melakukan generalisasi pada data yang belum pernah dilihat sebelumnya. Pendekatan ini penting karena memungkinkan kita untuk menghindari overfitting, yaitu keadaan di mana model terlalu sesuai dengan data pelatihan tetapi tidak mampu melakukan prediksi yang baik pada data baru. Dengan menguji model pada beberapa lipatan (fold) data yang berbeda, cross-validation memberikan perkiraan yang lebih akurat tentang seberapa baik model dapat memprediksi data yang belum dikenal. Ini membantu kita memastikan bahwa model yang dipilih memiliki kinerja yang baik secara umum dan dapat diterapkan pada data dunia nyata.

Heatmap untuk melihat fitur penting yang berkorelasi kuat dengan target



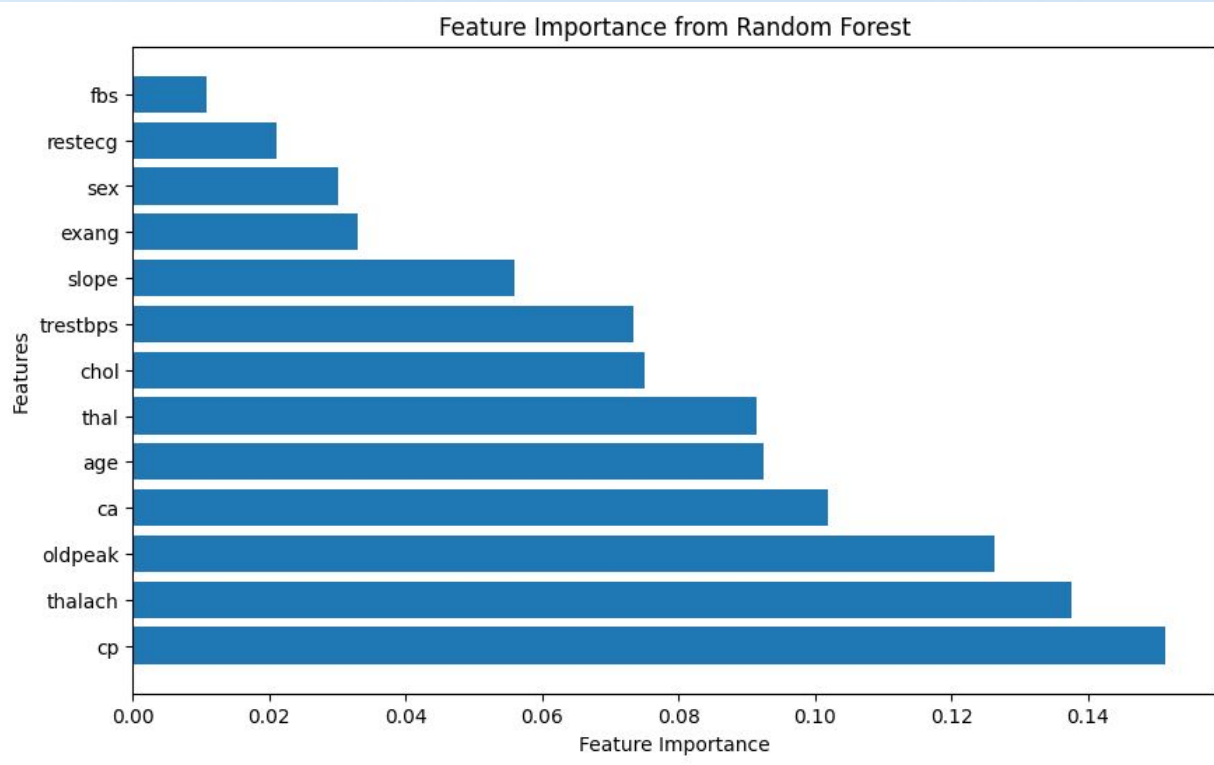
Berdasarkan correlation test, dapat diketahui bahwa ada hubungan positif antara cp (0.45), restecg (0.19), thalach (0.45), slope (0.36).

Korelasi yang paling kuat:

- positif dengan target cp, thalach, slope
- negatif dengan target exang, oldpeak, ca, thal

Seleksi fitur dengan Random Forest

Digunakan 6 fitur yang paling berkorelasi dengan target, untuk pemodelan ke-3. Yaitu fitur cp, thalach, oldpeak, ca, dan age

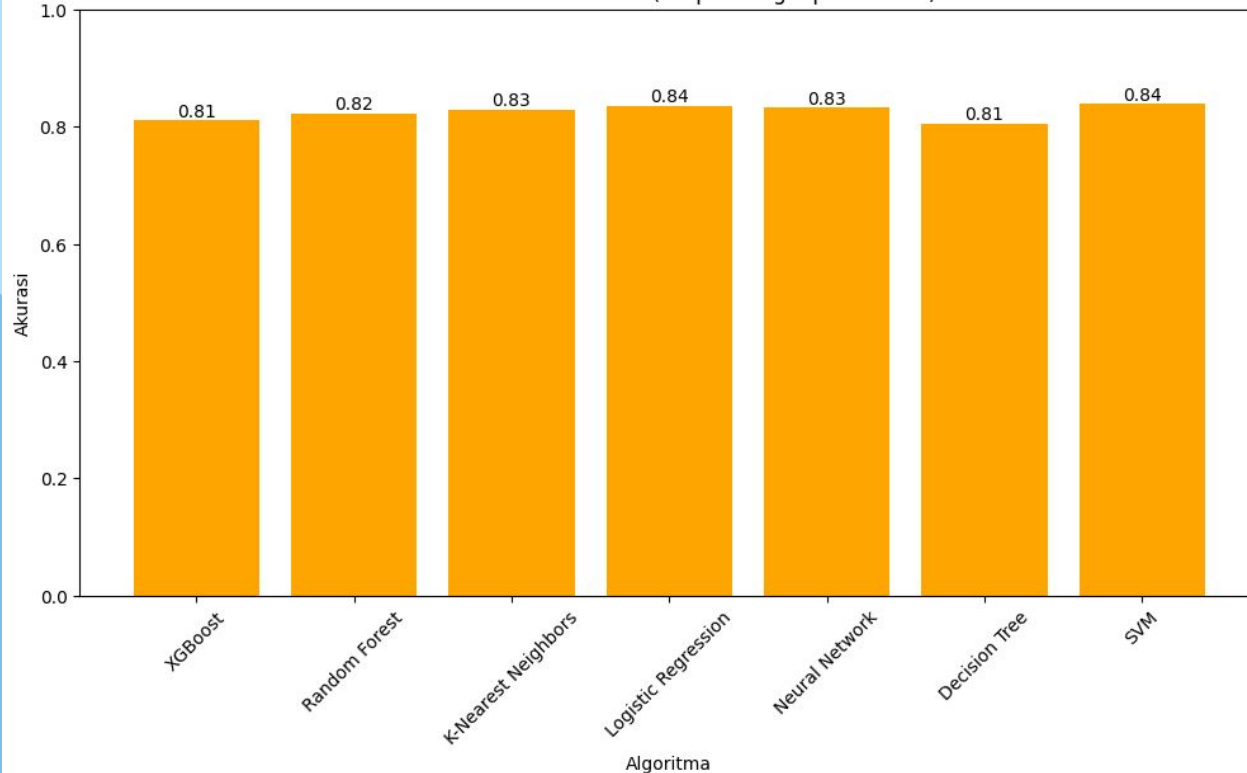


Fitur: cp, Feature Importance: 0.1514
Fitur: thalach, Feature Importance: 0.1376
Fitur: oldpeak, Feature Importance: 0.1262
Fitur: ca, Feature Importance: 0.1018
Fitur: age, Feature Importance: 0.0924
Fitur: thal, Feature Importance: 0.0913
Fitur: chol, Feature Importance: 0.0749
Fitur: trestbps, Feature Importance: 0.0733
Fitur: slope, Feature Importance: 0.0560
Fitur: exang, Feature Importance: 0.0329
Fitur: sex, Feature Importance: 0.0302
Fitur: restecg, Feature Importance: 0.0211
Fitur: fbs, Feature Importance: 0.0107

**Perbandingan
Performa tiap
algoritma dan
pengembangan
metodenya**

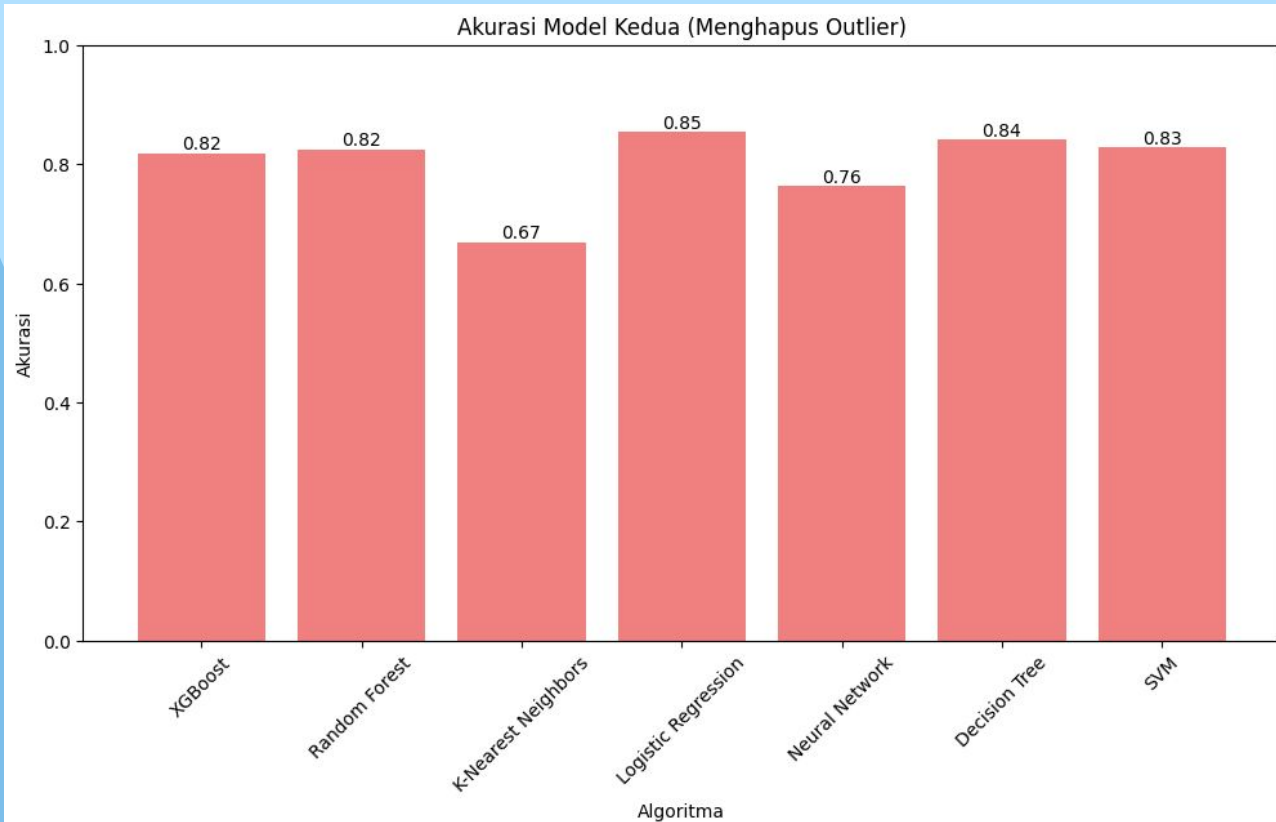
Perbandingan akurasi dari Pemodelan Pertama

Akurasi Model Pertama (Tanpa Menghapus Outlier)



Pada pemodelan yang pertama yang tidak menghapus outlier dan menggunakan seleksi fitur, diperoleh bahwa algoritma logistic regression dan SVM mampu memberi akurasi terbesar dibanding algoritma lainnya yaitu 84%. Sedangkan KNN dan Neural Network berhasil memperoleh akurasi 83%. Akurasi yang didapat dari pemodelan pertama ini cukup bagus, yang mana setiap algoritma mampu menghasilkan akurasi antara 81-84%.

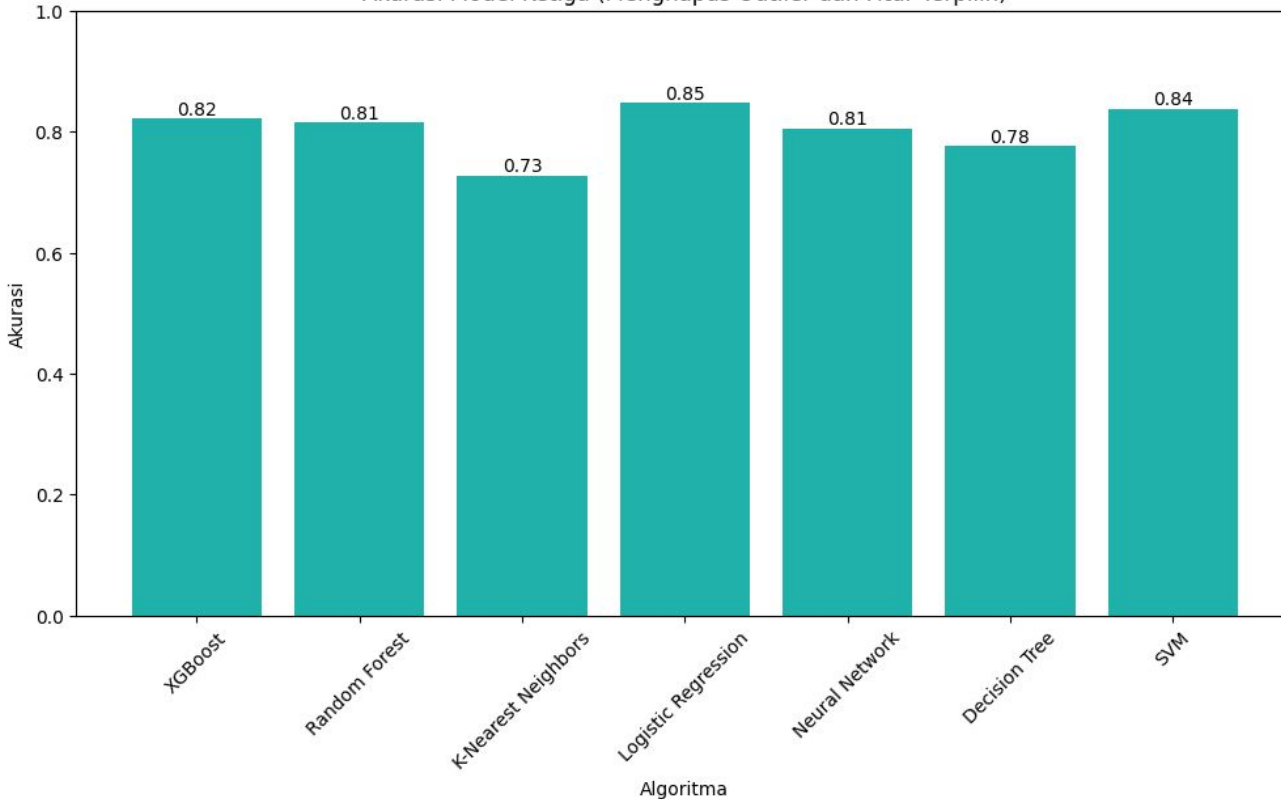
Perbandingan akurasi dari Pemodelan Pertama



Pemodelan kedua, menggunakan dataset, yang sebelumnya data outliernya dihapus terlebih dahulu. Yaitu menggunakan 308 record data. Meskipun akurasi yang dihasilkan dari KNN adalah 67% dan Neural Network 76%, pemodelan kedua ini berhasil memperoleh akurasi lebih besar dibanding pemodelan pertama. Yang mana algoritma Logistic Regression memperoleh akurasi sebesar 85%, dan Decision Tree sebesar 84%. Meskipun data yang dilatih cenderung lebih kecil dibanding pemodelan pertama, namun di algoritma LR dan DT memberi performa yang bagus.

Perbandingan akurasi dari Pemodelan Pertama

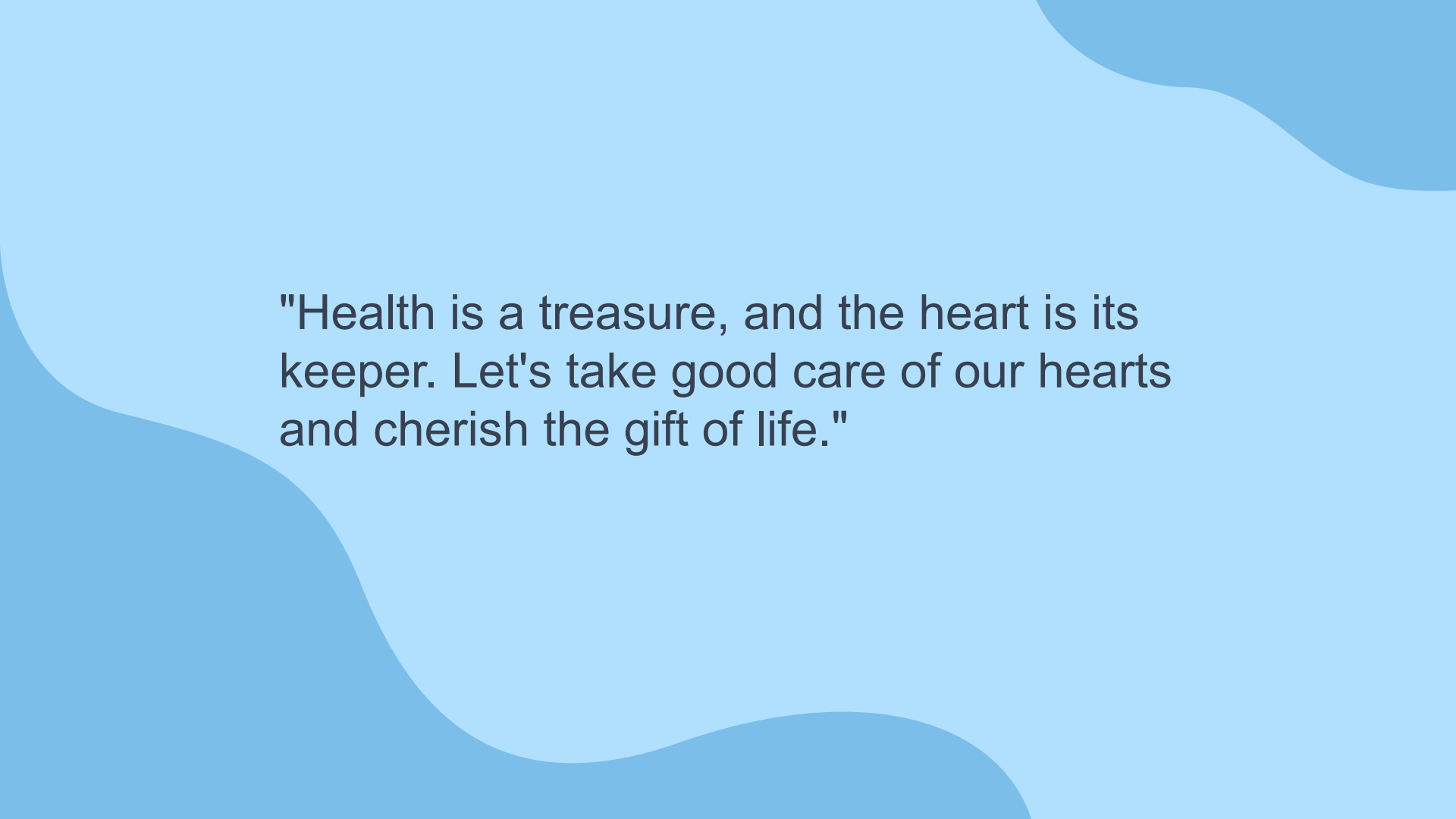
Akurasi Model Ketiga (Menghapus Outlier dan Fitur Terpilih)



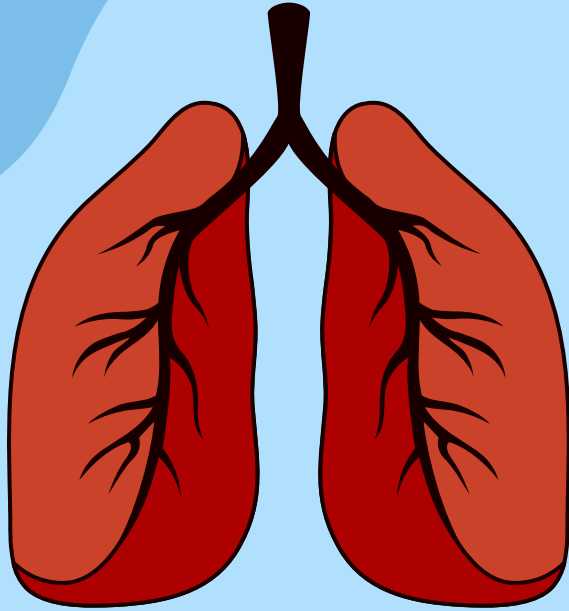
Pemodelan ketiga menggunakan data yang outliernya telah dihapus, dan hanya menggunakan 6 fitur saja. Yang mana fitur-fitur tersebut diperoleh dari Feature selection menggunakan Feature Importance. Namun meskipun demikian, dengan menggunakan data yang lebih minim dibanding pemodelan 1 dan 2, Logistic Regression tetap memberikan akurasi yang tinggi, sebesar 85%. Begitu juga SVM yang memperoleh akurasi yang cukup tinggi pula yaitu 84%. Namun akurasi terkecil yang diperoleh sebesar 73% dan 75%, dari algoritma KNN dan DT.

Kesimpulan

- Tiap algoritma memiliki cara kerja, kelebihan dan kekurangan sendiri terutama dalam menjalankan tugasnya.
- Teknik penyeimbangan data menggunakan SMOTE merupakan metode yang bagus dan tepat
- Begitu juga dengan pilihan menggunakan StandardScaler untuk Standarisasi data.
- Algoritma Logistic Regression dan SVM cenderung menunjukkan performa yang bagus dalam tiap model.
- Tiap metode yang diterapkan dalam 3 model, menunjukkan pengaruh yang cukup signifikan. Model ketiga yang dilakukan penghapusan outlier dan seleksi fitur memberikan prforma yang optimal. Bahkan sama dengan pemodelan kedua yang belum menerapkan feature selection. Yaitu pada algoritma Logistic Regression akurasi sebesar 85%. Apabila perfromanya sama, maka mungkin bisa dipilih pemodelan ketiga, menggunakan feature selection, karena fiturnya lebih sedikit tentunya daya dan waktu yang lebih dibutuhkan juga lebih sedikit.
- Akurasi terbesar diperoleh adalah 85%, apabila diinginkan mendapat akurasi yang lebih besar, mungkin selanjutnya bisa dicoba menggunakan algoritma lain yang lebih powerfull, atau mencoba membangun model stacking ensemble learning, vooting, baging atau lainnya. Hyperparameter Tunning juga mungkin bisa dilakukan untuk memaksimalkan akurasi, dengan menggunakan parameter yang tepat di algoritmanya. Atau juga menggunakan metode Feature Selection lainnya.

The background is a light blue color with several darker blue, wavy, organic shapes that resemble water or clouds. These shapes are positioned in the top right, bottom left, and bottom center areas, creating a layered, abstract effect.

"Health is a treasure, and the heart is its keeper. Let's take good care of our hearts and cherish the gift of life."



THANKS!

Link Colab:

<https://colab.research.google.com/drive/1D4aDirFhwfEy1HDewaJtVs2-B3xxo8BA?usp=sharing>

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**