

Machine Learning Analysis for Diabetes Prediction

Decision Tree vs SVM
Model Comparison

Nadhif Rif'at Rasendriya

Overview

Proyek ini bertujuan untuk memprediksi apakah seseorang menderita diabetes atau tidak berdasarkan beberapa fitur kesehatan, seperti usia, BMI, tekanan darah, dan kadar glukosa.

Model machine learning yang digunakan adalah:

- Decision Tree 🌳
- Support Vector Machine (SVM) ⚡

Dataset yang digunakan

- ✓ Data Training → Digunakan untuk melatih model
- ✓ Data Testing → Digunakan untuk mengevaluasi model

Fitur Dalam Dataset



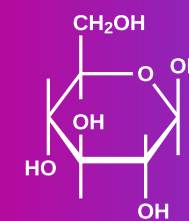
Age (Usia) → Umur pasien



BMI (Body Mass Index) → Indeks massa tubuh



Blood Pressure → Tekanan darah pasien



Glucose (Kadar Glukosa) →
Jumlah glukosa dalam darah



Diabetes (Label) →
1 jika pasien menderita diabetes, 0 jika tidak

Exploratory Data Analysis (EDA)

Sebelum membangun model, dilakukan analisis eksplorasi data (EDA) untuk memahami pola dalam dataset. Beberapa langkah yang dilakukan dalam EDA adalah:



a. Memeriksa Distribusi Data

- Melihat bagaimana data seperti usia, BMI, tekanan darah, dan glukosa tersebar.
- Menganalisis apakah ada outlier atau nilai ekstrim yang dapat mempengaruhi model.



b. Korelasi Antar Fitur

- Menganalisis hubungan antara BMI, tekanan darah, dan glukosa terhadap diabetes.
- Jika suatu fitur memiliki hubungan kuat dengan diabetes, maka fitur tersebut akan berpengaruh besar pada prediksi.



c. Data Cleaning

- Memeriksa missing values atau data yang hilang.
- Normalisasi atau standarisasi data jika diperlukan.

Full Code

https://github.com/nadhif-royal/DecisionTree_vs_SVM_DiabetesPrediction



Scan here

Datasets

Train

	Age	BMI	BloodPressure	Glucose	Diabetes
0	45	25.3	120	90	0
1	50	30.1	140	160	1
2	39	27.8	130	105	0
3	60	33.2	145	180	1
4	33	22.4	110	85	0
5	55	29.9	135	150	1
6	42	26.5	125	95	0
7	48	28.7	138	145	1
8	59	32.5	142	175	1
9	38	24.1	118	100	0

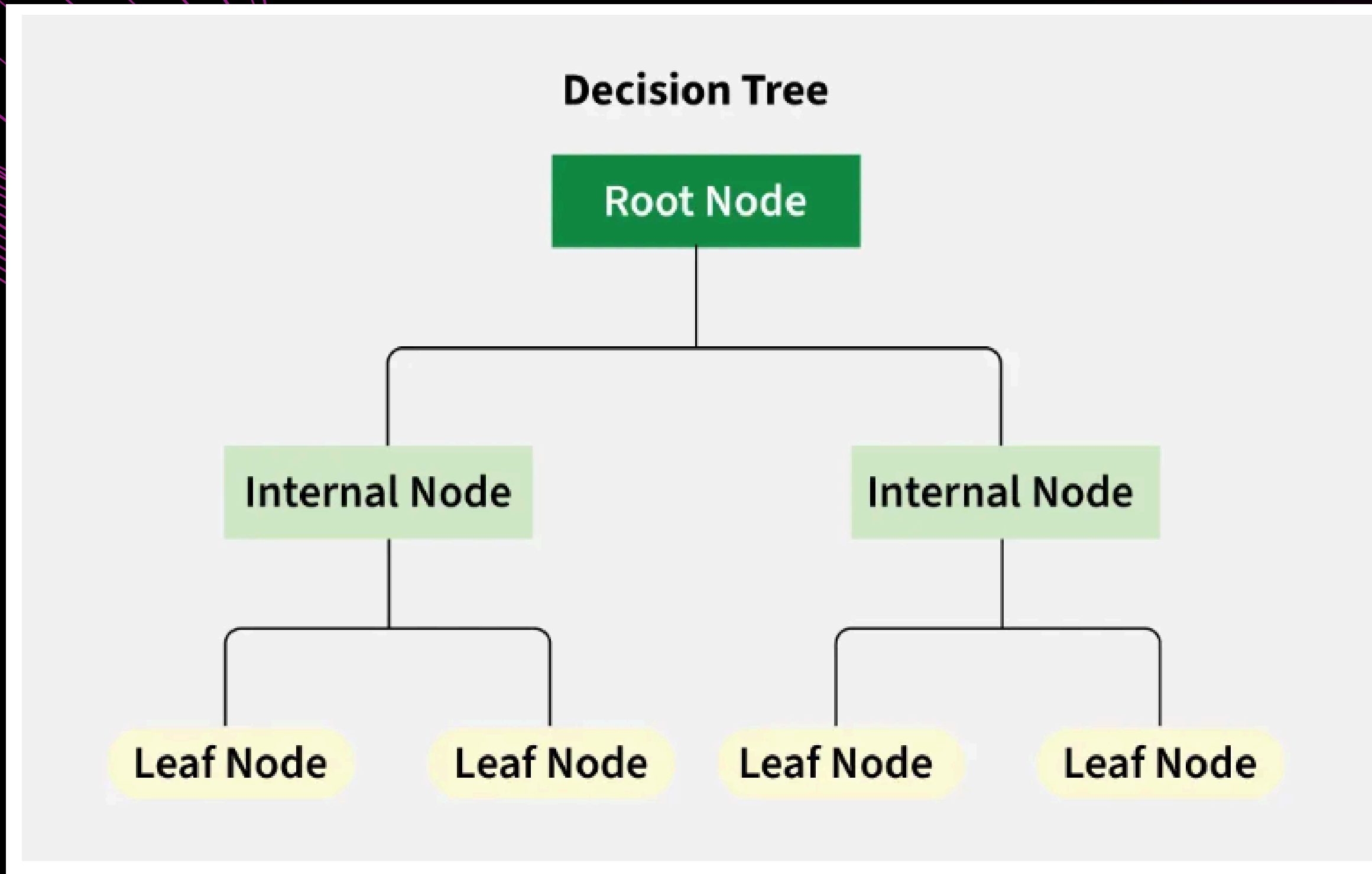
Test

	Age	BMI	BloodPressure	Glucose	Diabetes
0	52	29.5	137	155	1
1	41	25.7	122	98	0
2	36	22.9	115	88	0
3	58	31.2	140	170	1
4	47	27.3	128	110	0



Decision Tree

Overview



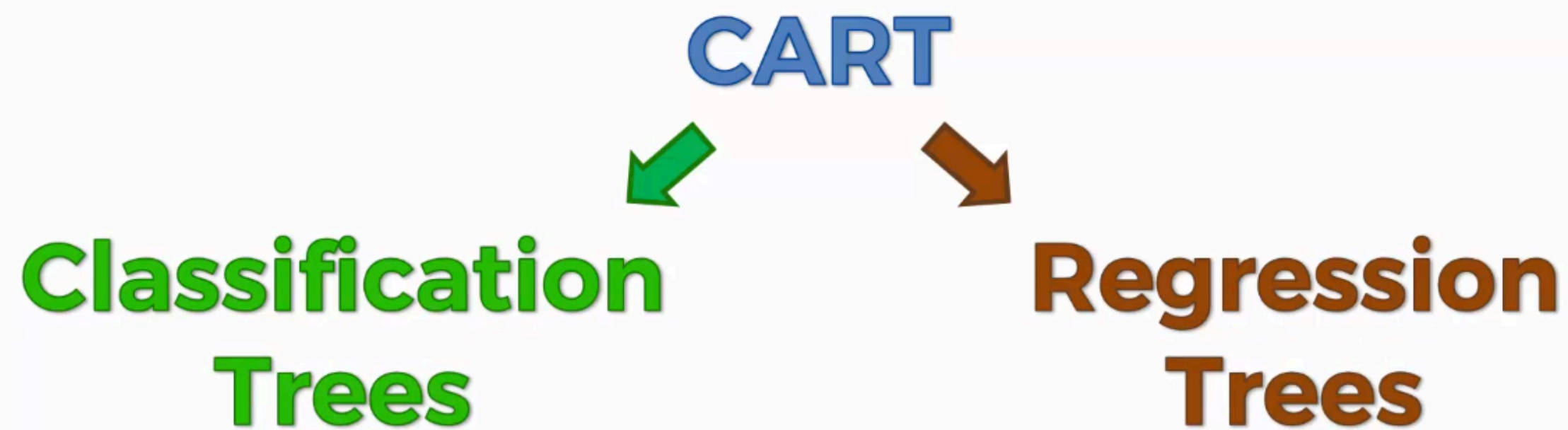
Decision tree adalah representasi grafis dari berbagai pilihan untuk memecahkan masalah dan menunjukkan bagaimana berbagai faktor terkait. Pohon keputusan memiliki struktur pohon hierarki yang dimulai dengan satu pertanyaan utama di bagian atas yang disebut simpul yang kemudian bercabang ke berbagai kemungkinan hasil yang berbeda.

Cara kerja Decision Tree sendiri mirip dengan konsep pohon faktor.

- Root Node adalah titik awal yang mewakili seluruh dataset.
- Branches: Ini adalah garis-garis yang menghubungkan node. Ini menunjukkan aliran dari satu keputusan ke keputusan lainnya.
- Internal Nodes adalah Titik-titik di mana keputusan dibuat berdasarkan fitur-fitur input.
- Leaf Nodes: Ini adalah node terminal di akhir cabang yang mewakili hasil akhir atau prediksi

Overview

Decision Tree Intuition



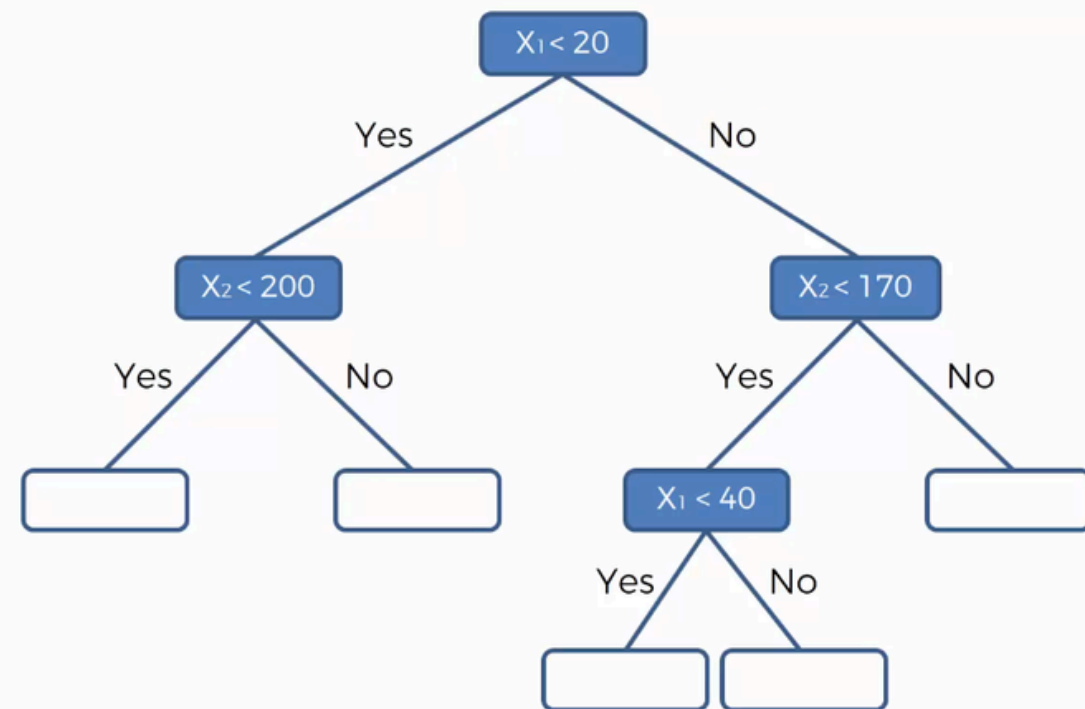
Decision Tree dibagi menjadi dua yaitu Klasifikasi dan Regresi.

- **Classification Trees:** Pohon ini dirancang untuk memprediksi hasil kategorikal yang berarti pohon ini mengklasifikasikan data ke dalam kelas-kelas yang berbeda. Pohon ini dapat menentukan apakah sebuah email adalah "spam" atau "bukan spam" berdasarkan berbagai fitur email.
- **Regression Trees:** Ini digunakan ketika variabel target bersifat kontinu. Ini memprediksi nilai numerik daripada kategori. Sebagai contoh, pohon regresi dapat memperkirakan harga rumah berdasarkan ukuran, lokasi, dan fitur lainnya.

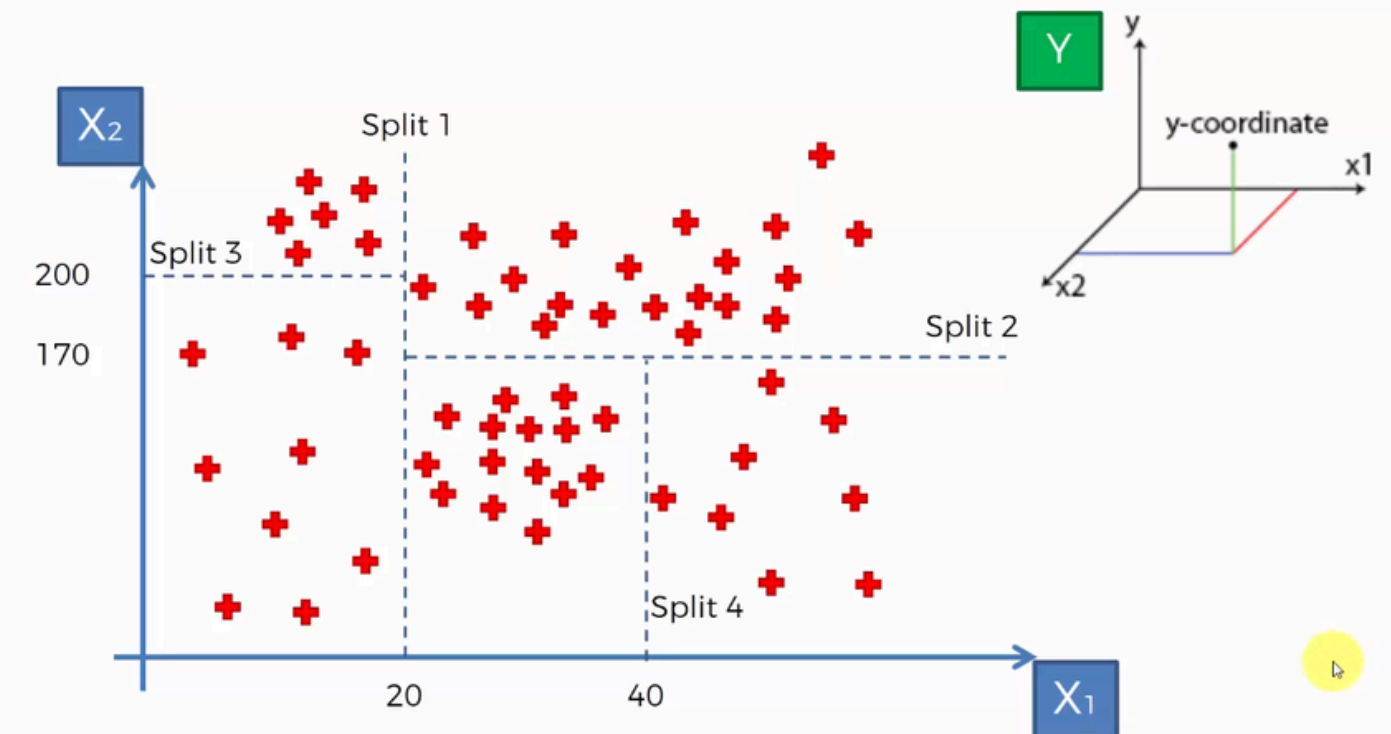
Intuition

Decision Tree

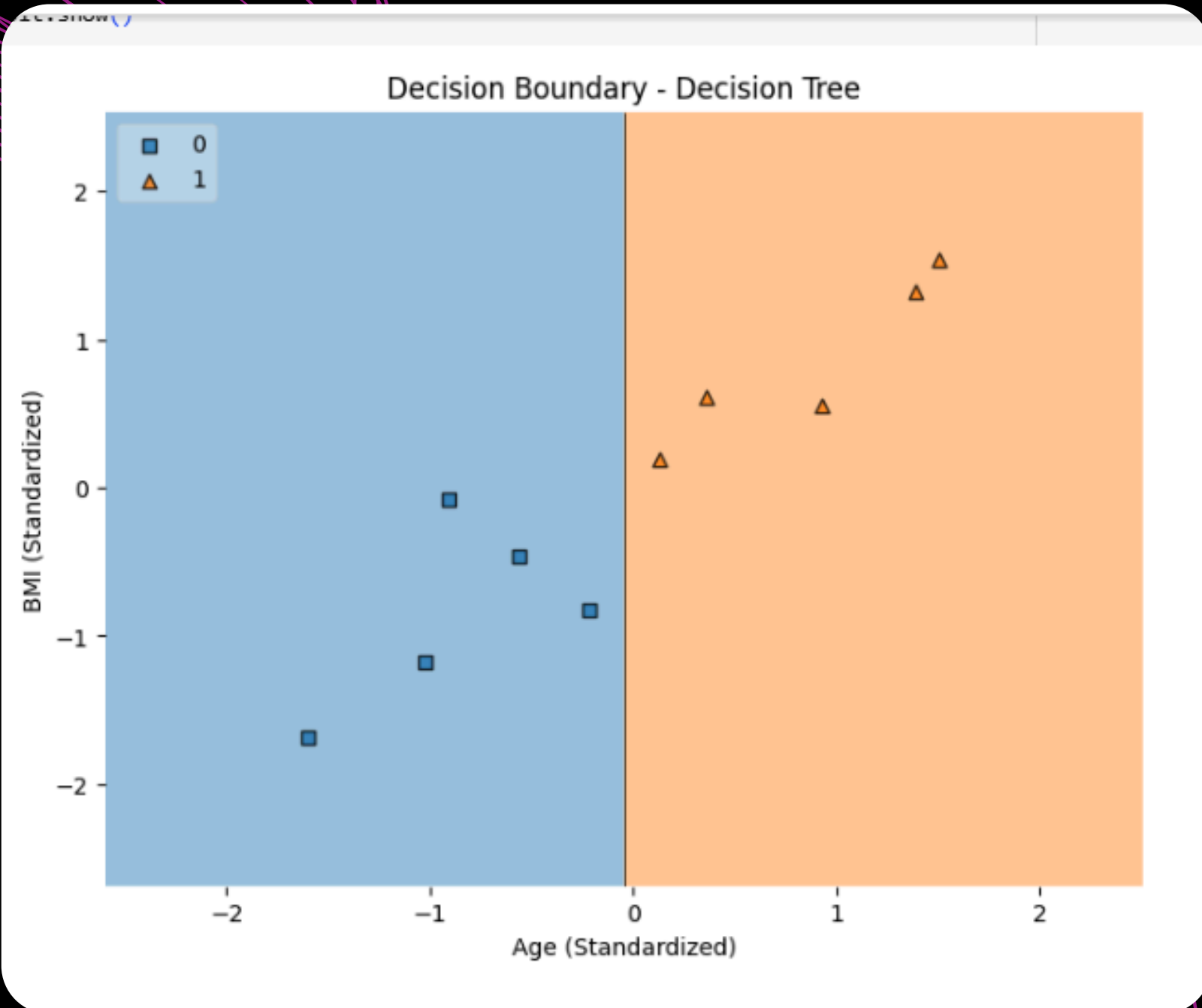
Decision Tree Intuition



Decision Tree Intuition



Visualization Decision Tree



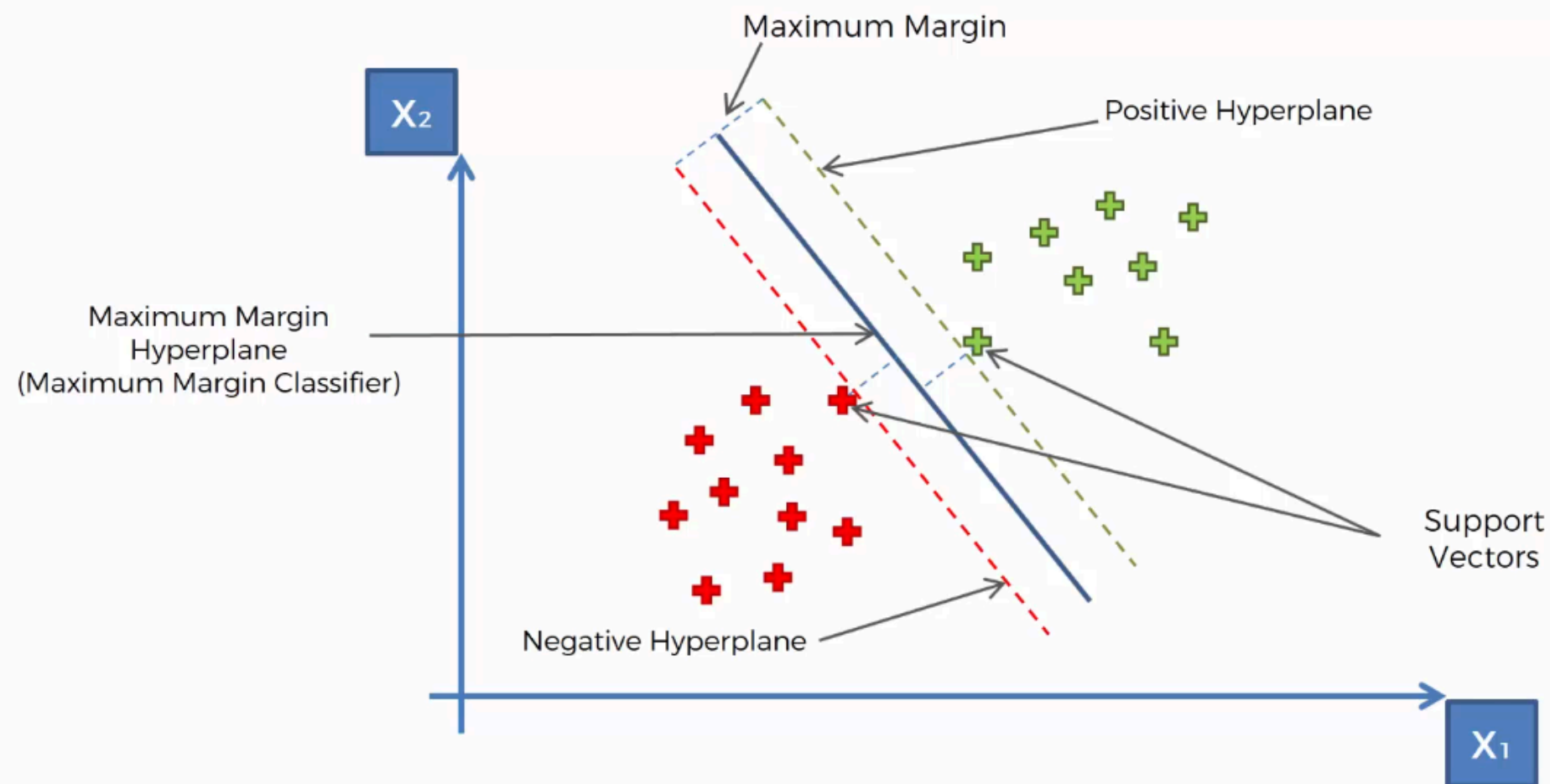
- Model Decision Tree membagi ruang fitur dengan garis tegas (vertikal) yang menunjukkan batas keputusan yang tidak terlalu kompleks.
- Model tampaknya memisahkan kelas berdasarkan nilai Age, dengan pemisahan utama di sekitar Age = 0 (standar).
- Hal ini menunjukkan bahwa Decision Tree lebih cenderung membuat keputusan berbasis aturan sederhana tanpa mempertimbangkan pola yang lebih kompleks dalam data.



Support Vector Machine (SVM)

Overview

Hyperplanes



Support Vector Machine (SVM) adalah algoritma pembelajaran mesin yang digunakan untuk klasifikasi dan regresi, tetapi lebih sering digunakan untuk klasifikasi. Model ini bekerja dengan mencari hyperplane terbaik yang memisahkan data ke dalam dua kelas yang berbeda.

Cara Kerja SVM:

SVM berusaha menemukan hyperplane optimal dengan margin terbesar antara dua kelas data. Hyperplane ini adalah garis pemisah (untuk data 2D) atau bidang pemisah (untuk data berdimensi lebih tinggi).

- Support Vectors: Titik-titik data terdekat dengan hyperplane yang menentukan batas margin.
- Maximal Margin: SVM mencoba membuat jarak antara hyperplane dan support vectors sebesar mungkin agar model lebih generalisasi.

Jika data tidak bisa dipisahkan secara linear, SVM menggunakan Kernel Trick untuk memetakan data ke dimensi yang lebih tinggi agar bisa dipisahkan dengan lebih baik.

What's So Special About SVM?

Yang membuat svm unik dibandingkan cara kerja model machine learning lainnya adalah, ketika kita memberikan data set untuk melatih model ini, alih alih melihat dataset ini secara general/standar, model ini justru mencari data yang paling unik yang mendekati hasil / tes yang di inginkan.

Jadi ketika kita memberi suatu data untuk dia menentukan data tersebut masuk ke golongan mana, maka model ini akan otomatis mencari data tersebut paling mirip dengan golongan yang mana.

4 hal utama yang membuat model ini memiliki kinerja yang jauh lebih baik adalah:

- ◆ **Berfokus pada Support Vectors**

Alih-alih mempertimbangkan seluruh data secara merata seperti model lain (misalnya k-NN yang melihat semua tetangga terdekat), SVM hanya menggunakan "support vectors"—yakni data yang berada di batas margin antar kelas.

- ◆ **Mencari Hyperplane Optimal**

SVM mencari hyperplane terbaik dengan margin terbesar, sehingga modelnya lebih generalisasi dan tidak mudah overfitting.

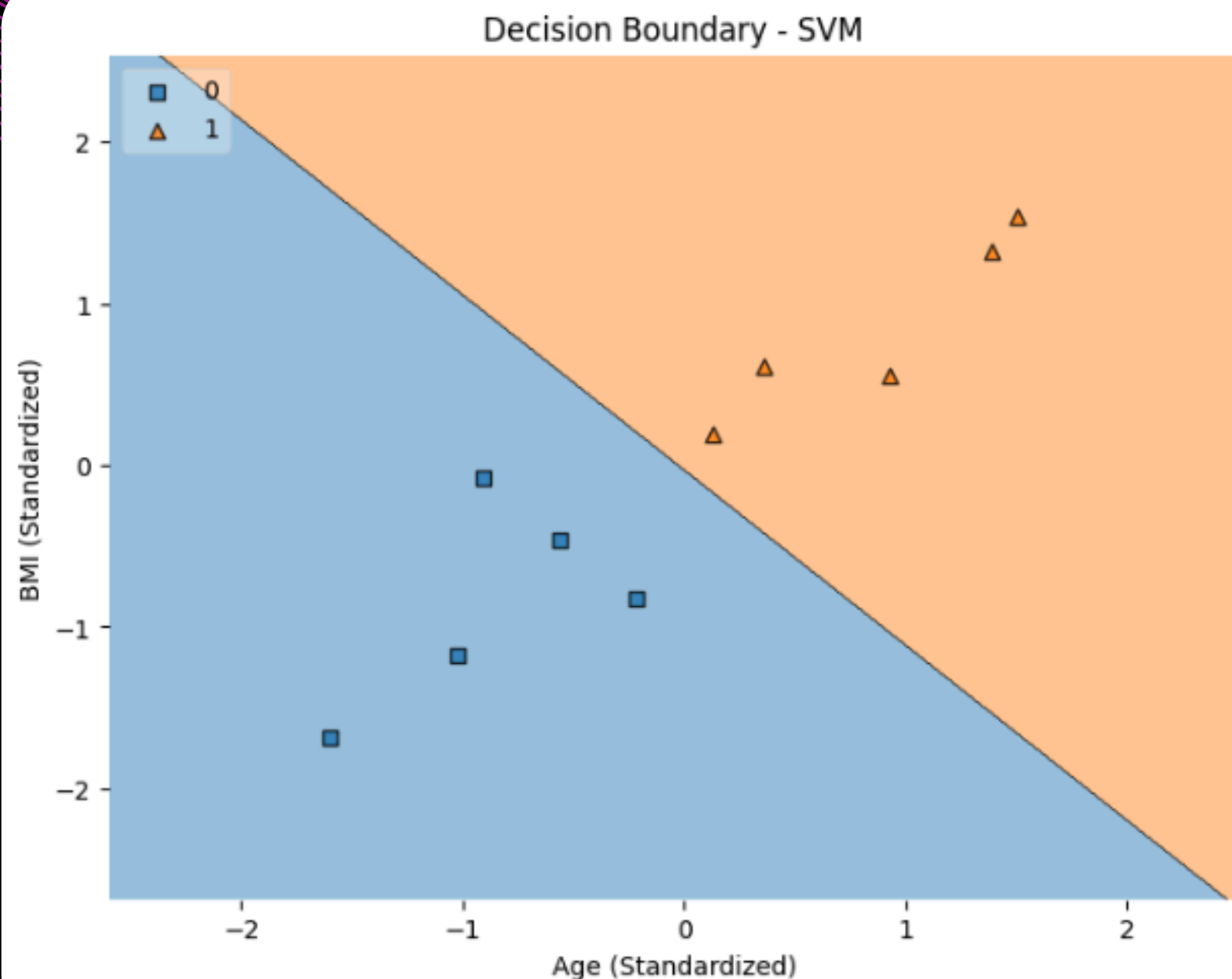
- ◆ **Bekerja dengan Baik dalam Dimensi Tinggi**

Ketika data tidak bisa dipisahkan secara linear, SVM menggunakan Kernel Trick untuk memproyeksikan data ke dimensi yang lebih tinggi agar lebih mudah dipisahkan.

- ◆ **Cocok untuk Data dengan Struktur Kompleks**

Dibandingkan dengan Decision Tree yang cenderung membuat aturan berdasarkan keseluruhan dataset atau Random Forest yang menggabungkan banyak pohon keputusan, SVM lebih selektif dalam memilih fitur penting dari data.

Visualization SVM (Support Vector Machine)



Grafik ini menampilkan Decision Boundary (Batas Keputusan) dari model SVM (Support Vector Machine) yang digunakan untuk membedakan dua kelas berdasarkan Age (Standardized) dan BMI (Standardized).

- SVM pada grafik ini menunjukkan keputusan linier yang cukup baik dalam membagi dua kelas berdasarkan fitur Age dan BMI.
- Batas keputusan diagonal menunjukkan bahwa kedua fitur memiliki kontribusi dalam pemisahan kelas.
- Jika dataset lebih kompleks, bisa mempertimbangkan kernel SVM agar hasilnya lebih optimal.



Comparison (Analisis Perbandingan)

The Result

Decision Tree

- ✓ Akurasi: 0.80
- ✓ Precision: 0.67
- ✓ Recall: 1.00
- ✓ F1 Score: 0.80
- ✓ Mean Squared Error: 0.20
- ✓ R² Score: 0.17

SVM

- ✓ Akurasi: 1.00
- ✓ Precision: 1.00
- ✓ Recall: 1.00
- ✓ F1 Score: 1.00
- ✓ Mean Squared Error: 0.00
- ✓ R² Score: 1.00

Diambil
berdasarkan hasil
comparison /
evaluasi model
pada file.ipynb

Perbandingan Model Decision Tree vs SVM

Berdasarkan hasil evaluasi, berikut adalah perbandingan antara Decision Tree dan Support Vector Machine (SVM) dalam memprediksi diabetes:

Metode	Akurasi	Precision	Recall	F1 Score	MSE	R ² Score
Decision Tree	0.80	0.67	1.00	0.80	0.20	0.17
SVM	1.00	1.00	1.00	1.00	0.00	1.00

Analisis Perbandingan

Akurasi, Precision, Recall, & F1 Score

- SVM memiliki akurasi 1.00 (100%), Precision 1.00, Recall 1.00, dan F1 Score 1.00, yang berarti model ini mampu mengklasifikasikan semua data dengan sempurna tanpa kesalahan.
- Decision Tree hanya mencapai akurasi 80%, Precision 0.67, Recall 1.00, dan F1 Score 0.80, yang berarti model ini masih memiliki beberapa kesalahan dalam prediksi positif.

Mean Squared Error (MSE)

- SVM memiliki MSE 0.00, yang berarti tidak ada kesalahan dalam prediksi.
- Decision Tree memiliki MSE 0.20, yang menunjukkan masih ada tingkat kesalahan dalam model ini.

R² Score

- SVM memiliki nilai R² Score = 1.00, yang berarti model ini mampu menjelaskan 100% variabilitas dalam data.
- Decision Tree memiliki R² Score = 0.17, yang cukup rendah, menunjukkan bahwa model ini kurang baik dalam menjelaskan variasi data.

Note: Hasil akan lebih akurat jika menggunakan Hyperparameter Tuning untuk menghindari adanya Overfitting.

Kesimpulan

- ◆ SVM adalah model terbaik untuk dataset ini, karena memberikan hasil sempurna dengan akurasi 1.00, Precision 1.00, Recall 1.00, F1 Score 1.00, serta R^2 Score 1.00. Model ini tidak memiliki kesalahan prediksi ($MSE = 0.00$), sehingga sangat andal untuk klasifikasi dalam kasus ini.
- ◆ Decision Tree masih cukup baik dengan akurasi 80%, Precision 0.67, Recall 1.00, dan F1 Score 0.80. Namun, performanya lebih rendah dibandingkan dengan SVM, terutama karena precision yang lebih rendah. Model ini bisa menjadi pilihan jika ingin interpretasi yang lebih mudah, tetapi kurang optimal dalam kasus ini karena masih memiliki kesalahan prediksi ($MSE = 0.20$) dan R^2 Score yang rendah (0.17).



Terimakasih



[linkedin.com/in/royalnadhif50/](https://www.linkedin.com/in/royalnadhif50/)