



Randi Nandika Danendra

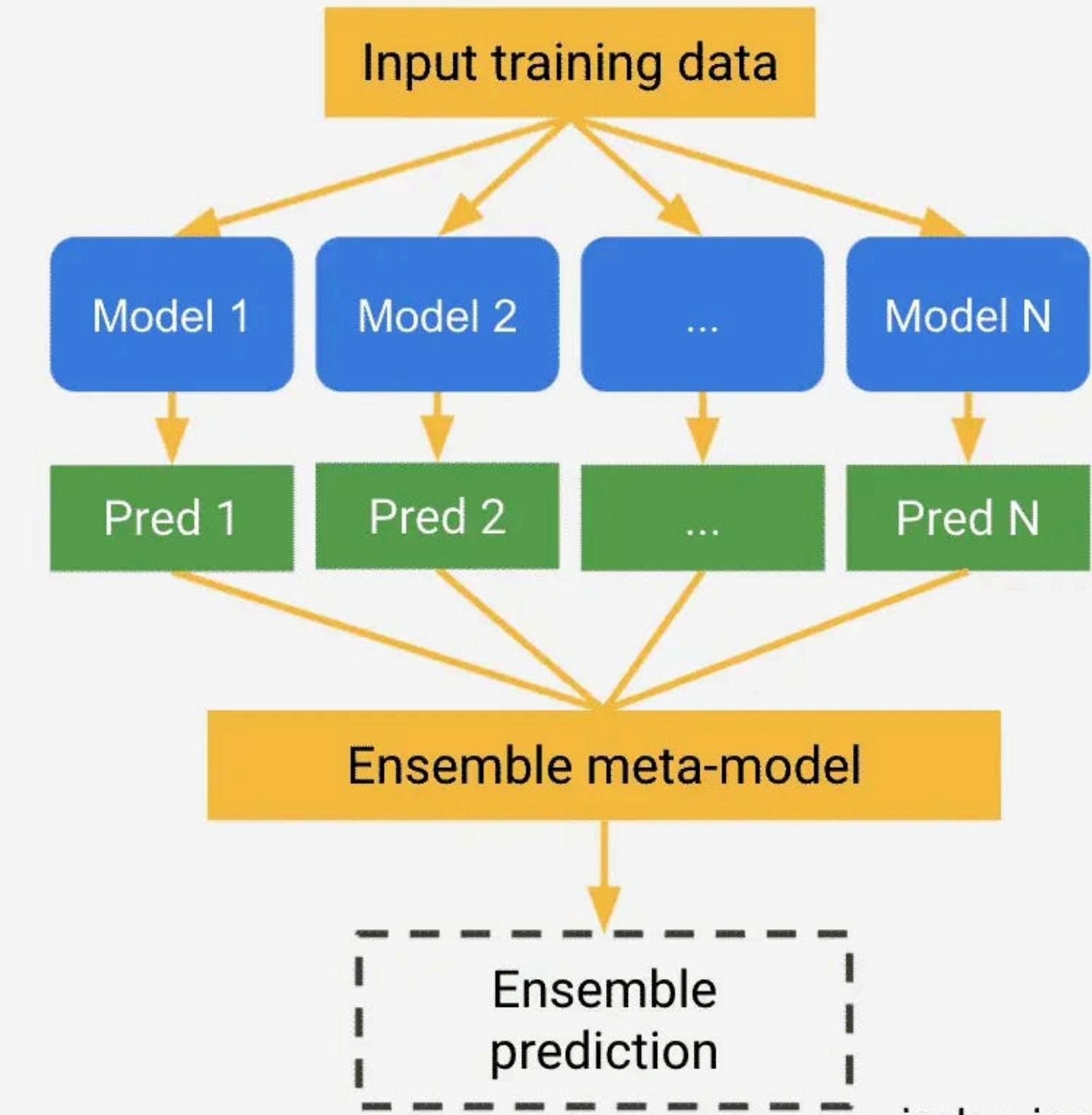
Ensemble Learning

Iris Dataset dan
HeartAttack Dataset

3324600013 - SDT A



Ensemble Learning



jcchouinard.



01

Ensemble Learning

02

Bootstrap Aggregating

The background features the Larana, Inc. logo, which includes a stylized book icon and the company name. Below the logo is a collage of many diverse people's faces, suggesting a global or diverse audience.

Introduction

Ensemble Learning adalah teknik yang menggabungkan beberapa model (weak learners) untuk membentuk satu model kuat (strong learner). Tujuannya: meningkatkan akurasi, stabilitas, dan generalization model.

Bagging adalah singkatan dari Bootstrap Aggregating, yaitu salah satu metode ensemble yang bekerja dengan mengurangi variansi model (biasanya model yang mudah overfitting seperti decision tree).

Langkah-langkah Bagging:

Tujuan

Tujuan	Penjelasan
Mengurangi variansi	Model seperti Decision Tree sangat sensitif terhadap perubahan kecil di data. Dengan bagging, hasil jadi lebih stabil.
Meningkatkan akurasi	Gabungan model sering kali lebih akurat daripada satu model tunggal.
Lebih tahan terhadap overfitting	Karena model dilatih pada subset data acak, tidak mudah terlalu “menempel” pada noise.

LEARN MORE

www.reallygreatsite.com





Data Iris

Membandingkan base classifier (Decision Tree) dengan dua model bagging classifier (Random Forest & Extra Trees), lalu menganalisis performa mereka menggunakan Confusion Matrix, ROC/AUC, atau Learning Curve.

Confusion
Matrix

ROC/AUC

Learning Curve

Preprocessing

Komponen	Nama Variabel	Isi / Peran	Keterangan
Fitur (input)	X	4 kolom numerik (SepalLength, SepalWidth, PetalLength, PetalWidth)	Data yang digunakan model untuk belajar
Target (output)	y	Label numerik hasil encoding dari Species	Nilai yang akan diprediksi oleh model
Data latih	X_train, y_train	80% data	Digunakan untuk melatih model
Data uji	X_test, y_test	20% data	Digunakan untuk menguji model

```
# Tampilkan contoh isi data training & testing
print("\nContoh 5 baris data training:")
print(X_train.head())

print("\nContoh label training:")
print(y_train[:10])
```

Fitur (X) setelah preprocessing:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

Target (y) setelah label encoding:
[0 0 0 0 0 0 0 0 0]

Ukuran data:

X_train: (120, 4)
X_test : (30, 4)
y_train: (120,)
y_test : (30,)

Contoh 5 baris data training:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
8	4.4	2.9	1.4	0.2
106	4.9	2.5	4.5	1.7
76	6.8	2.8	4.8	1.4
9	4.9	3.1	1.5	0.1
89	5.5	2.5	4.0	1.3

Contoh label training:
[0 2 1 0 1 2 1 2 2 2]

LEARN MORE

inisialisasi Model

• • • •
• • • •
• • • •
• • • •
• • • •

```
# === 6. Inisialisasi Model ===
models = {
    'Decision Tree': DecisionTreeClassifier(random_state=42),
    'Random Forest': RandomForestClassifier(n_estimators=100, random_state=42),
    'Extra Trees': ExtraTreesClassifier(n_estimators=100, random_state=42)
}
```

Baris kode ini adalah inti dari perbandingan model bagging:

- Decision Tree → baseline model tunggal.
- Random Forest → Bagging klasik (menggabungkan banyak pohon).
- Extra Trees → Bagging dengan tingkat randomisasi lebih tinggi.

LEARN MORE

www.reallygreatsite.com

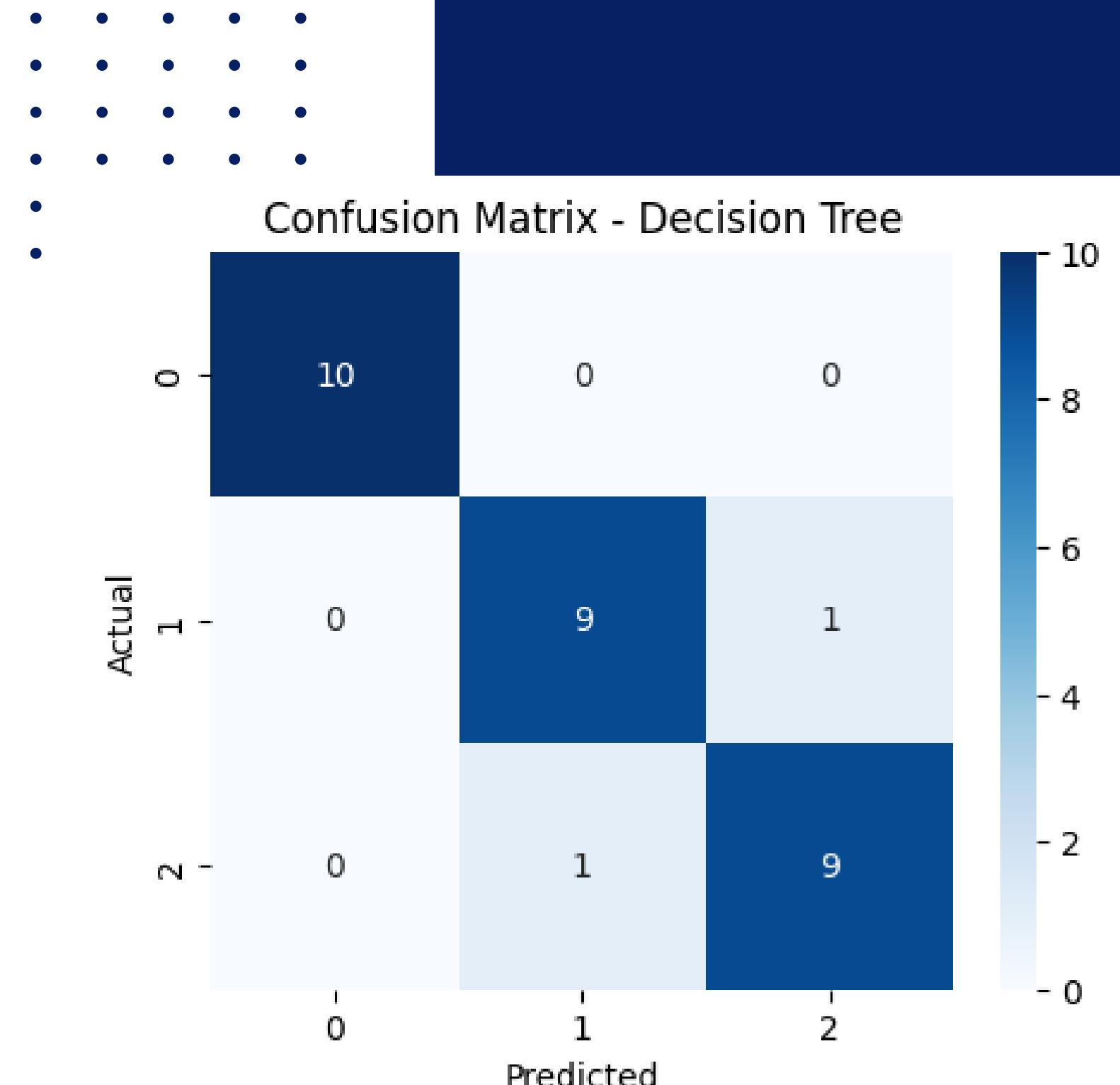
Evaluasi Model Matrix Confusi



Decision Tree

```
...  
===== Model: Decision Tree =====  
precision recall f1-score support  
0 1.00 1.00 1.00 10  
1 0.90 0.90 0.90 10  
2 0.90 0.90 0.90 10  
  
accuracy 0.93  
macro avg 0.93 0.93 0.93 30  
weighted avg 0.93 0.93 0.93 30
```

- Hampir sempurna, hanya 2 kesalahan kecil (kelas 1 salah satu kali jadi 2, dan sebaliknya).
- Akurasi total: 93%

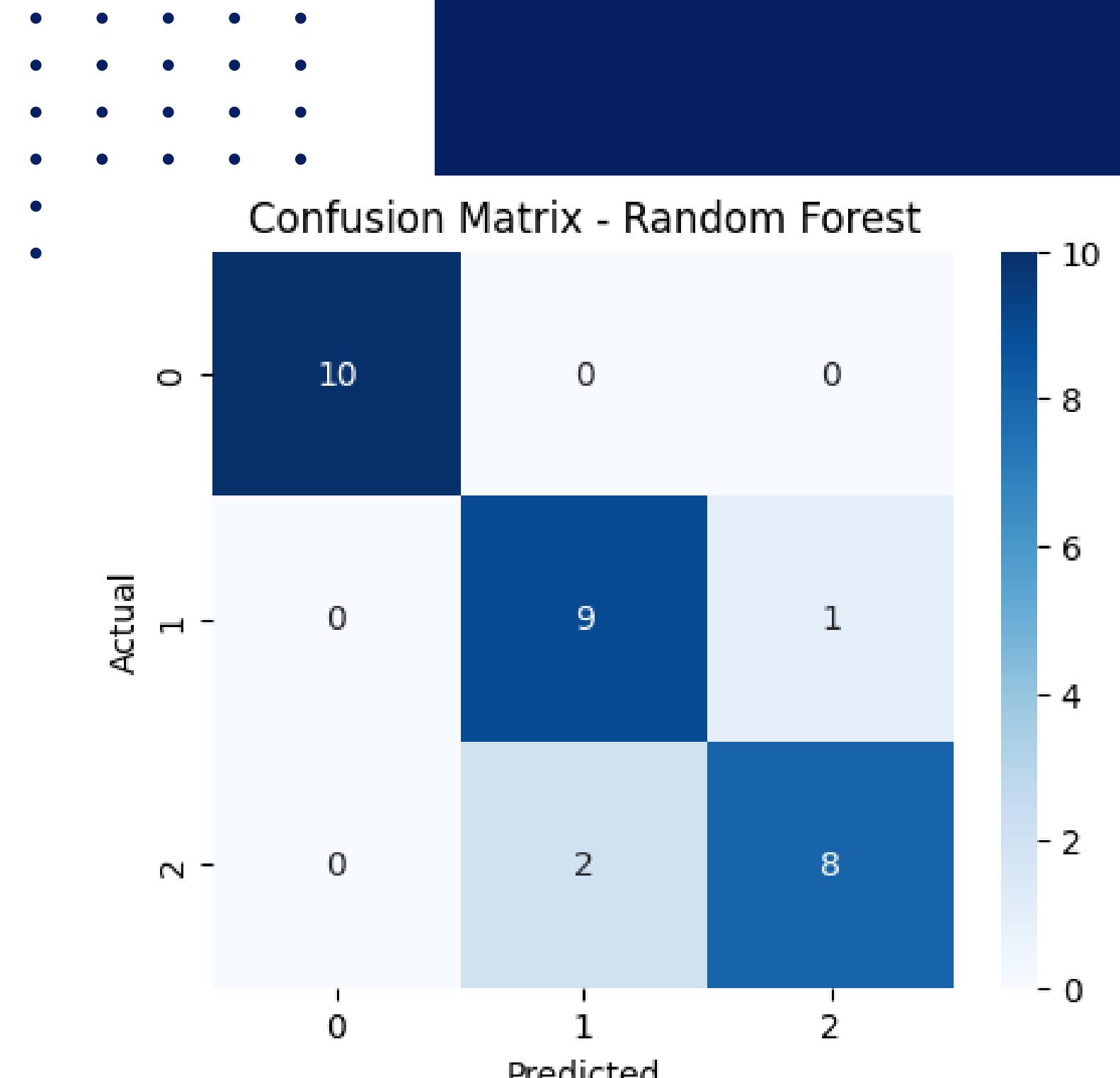


[LEARN MORE](#)

Random Forest

Model: Random Forest				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	10
1	0.82	0.90	0.86	10
2	0.89	0.80	0.84	10
accuracy			0.90	30
macro avg	0.90	0.90	0.90	30
weighted avg	0.90	0.90	0.90	30

- Masih sangat baik, tapi ada sedikit penurunan akurasi dibanding Decision Tree (kesalahan kelas 2 sedikit meningkat).
- Akurasi total: 90%

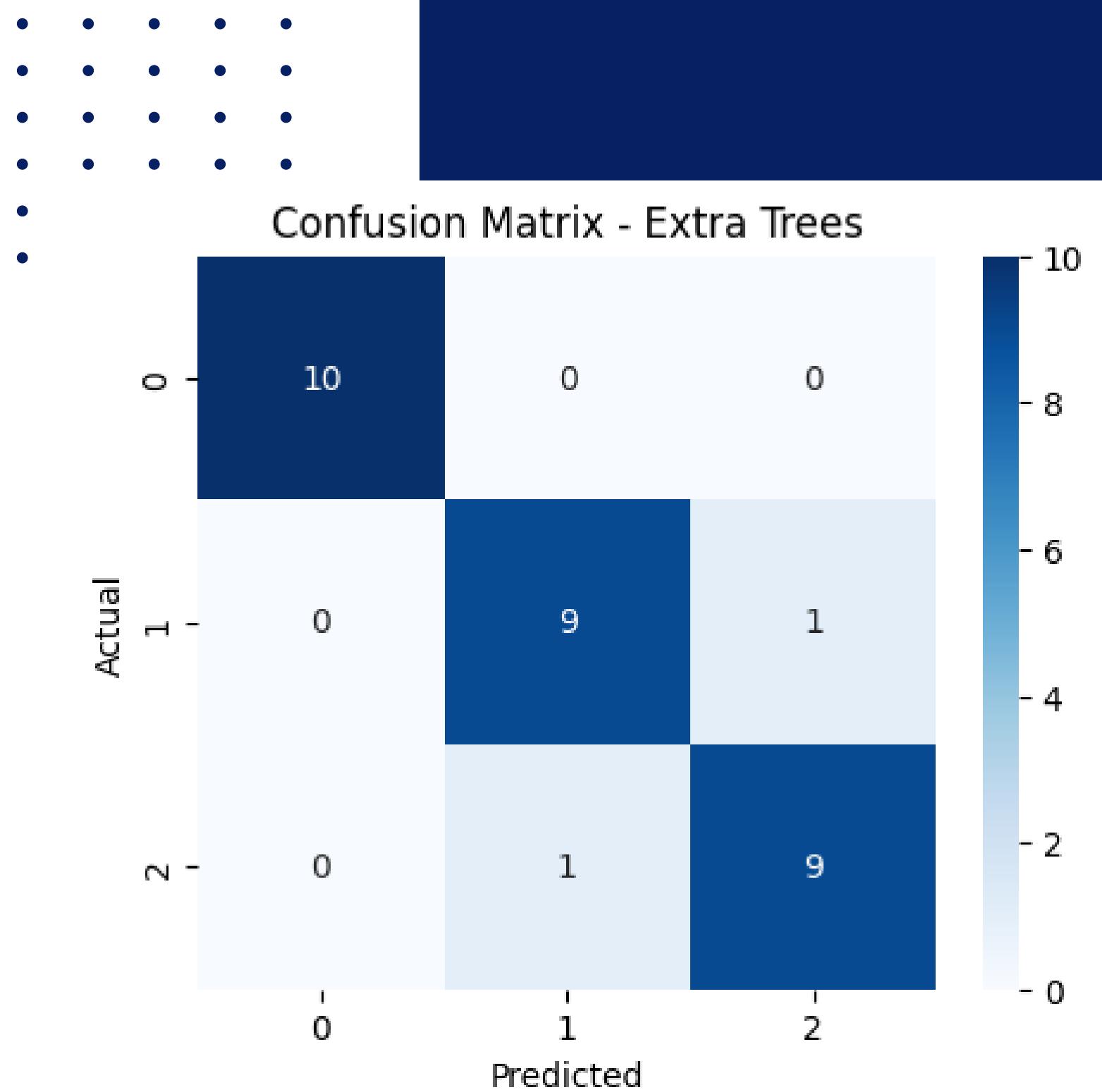


LEARN MORE

Extra Trees

Model: Extra Trees				
	precision	recall	f1-score	support
0	1.00	1.00	1.00	10
1	0.90	0.90	0.90	10
2	0.90	0.90	0.90	10
accuracy			0.93	30
macro avg	0.93	0.93	0.93	30
weighted avg	0.93	0.93	0.93	30

- Hasil hampir identik dengan Decision Tree, akurasi kembali 93%.

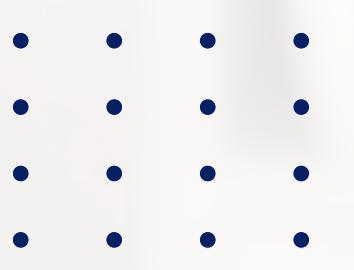


[LEARN MORE](#)

Perbandingan

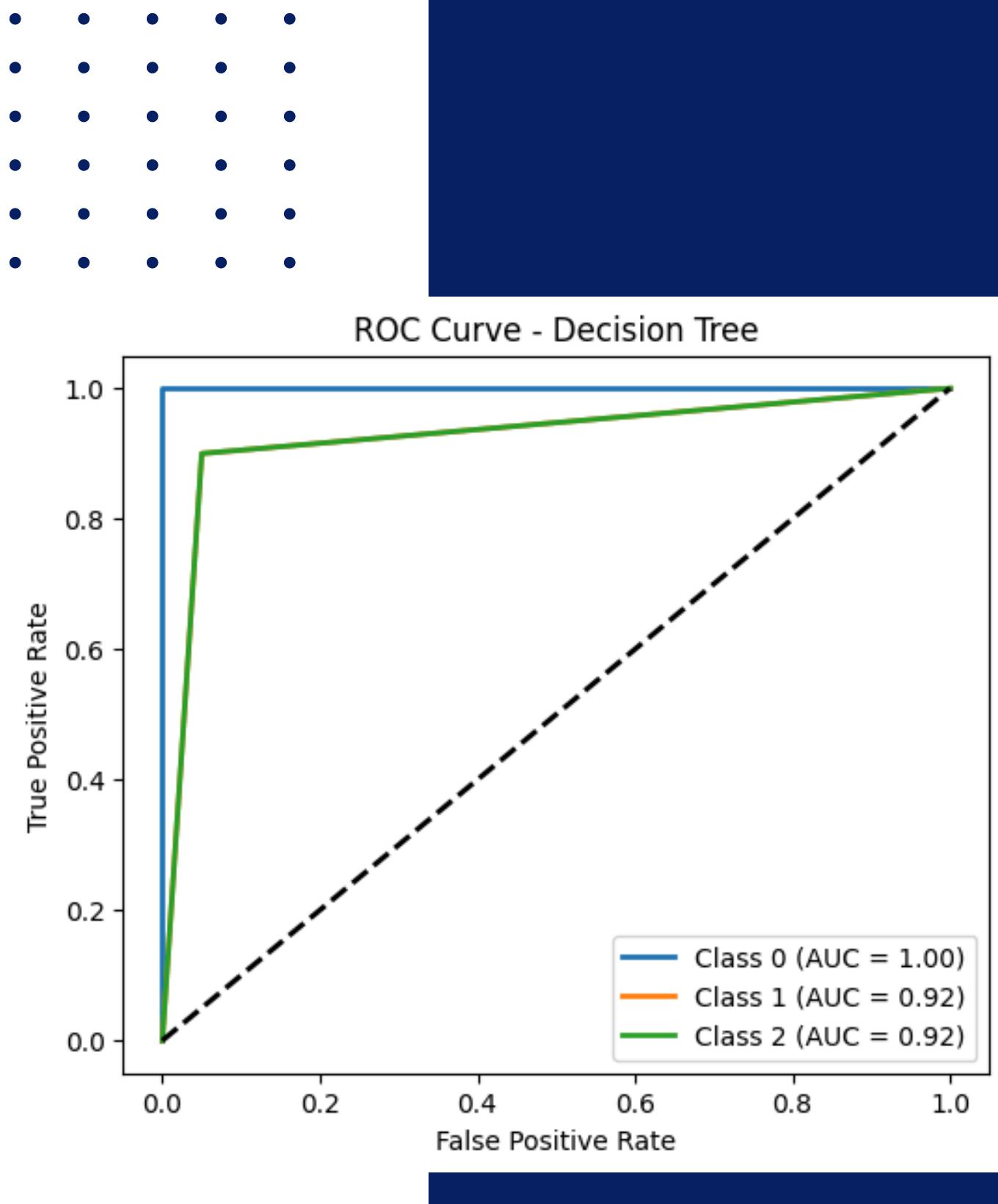
Model	Jenis	Teknik	Akurasi	Catatan
Decision Tree	Base Model	Single Tree	93%	Akurat tapi rentan overfitting
Random Forest	Bagging	Banyak pohon (bootstrap sample)	90%	Lebih stabil, sedikit menurun karena agregasi
Extra Trees	Bagging (lebih acak)	Banyak pohon (split acak)	93%	Cepat dan stabil, variansi rendah

Evaluasi Model ROC AUC



Decision Tree

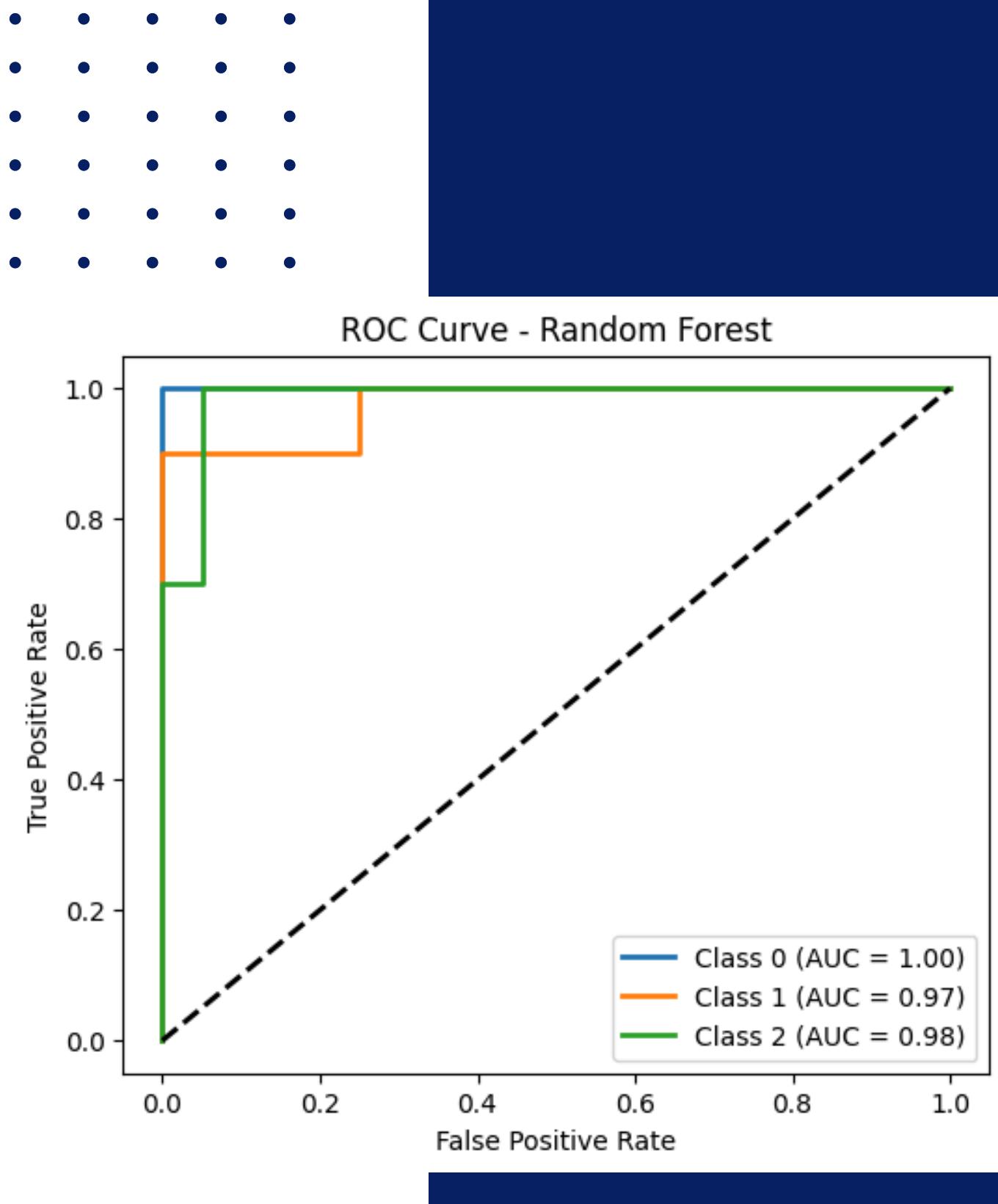
- Decision Tree mampu membedakan Class 0 (Iris Setosa) secara sempurna ($AUC = 1.00$).
- Namun untuk Class 1 dan 2 (Versicolor dan Virginica), AUC hanya 0.92 – menandakan model kurang optimal membedakan dua kelas yang bentuknya mirip.
- Pola ROC terlihat lebih tajam di awal lalu melandai, yang artinya sensitivitasnya bagus di awal threshold tapi tidak terlalu stabil di tengah.



[LEARN MORE](#)

Random Forest

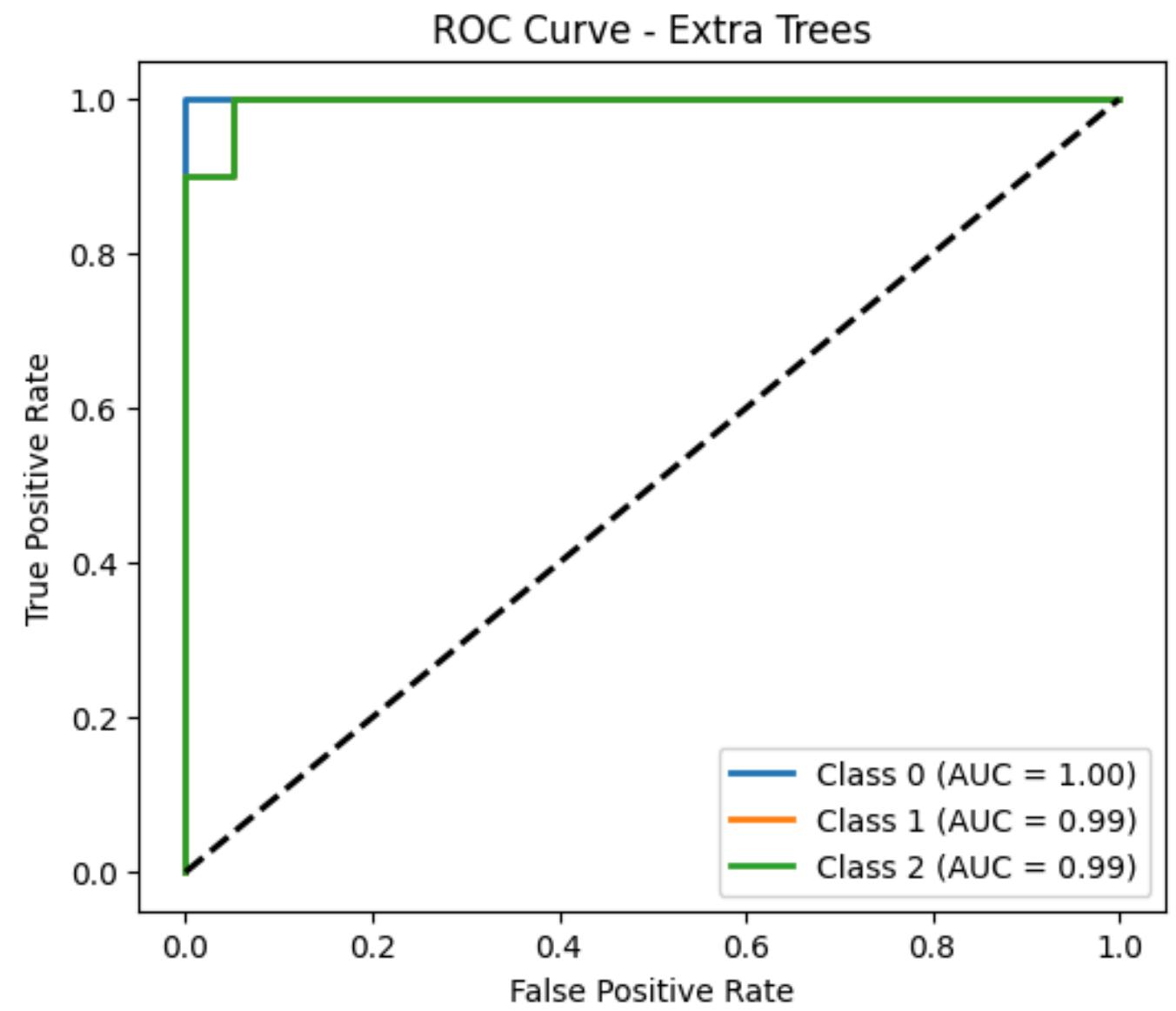
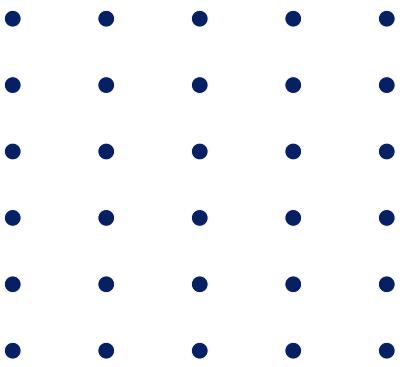
- Semua kelas memiliki AUC di atas 0.95, menunjukkan performansi yang sangat stabil dan kuat.
- Performa meningkat dibanding Decision Tree di Class 1 dan 2 (dari $0.92 \rightarrow 0.97 \& 0.98$).
- Ini membuktikan efek bagging mengurangi variansi dan meningkatkan generalisasi.
- Kurva ROC sangat dekat dengan sisi kiri atas grafik (ideal).



[LEARN MORE](#)

Extra Trees

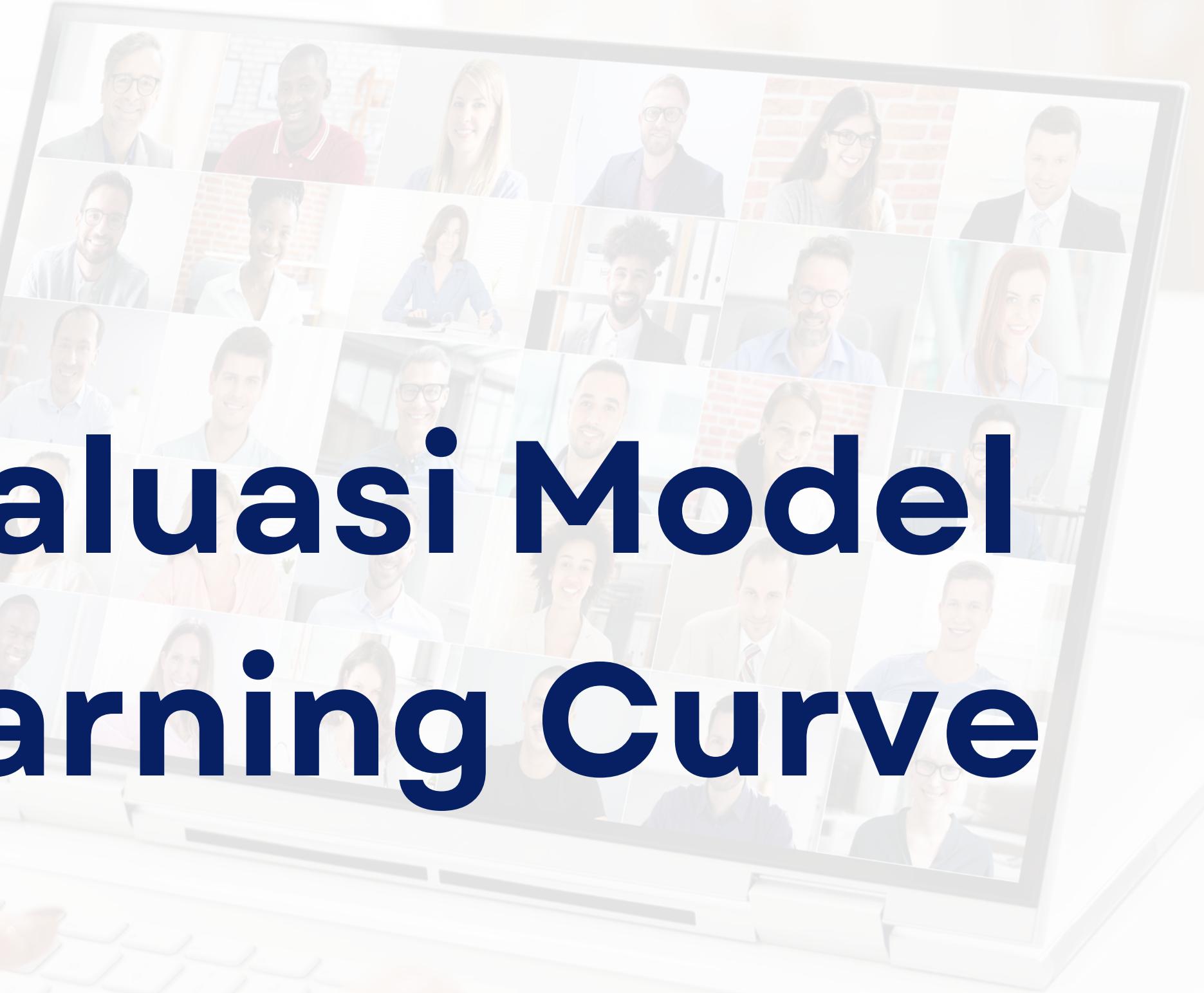
- Extra Trees menunjukkan hasil paling sempurna dan stabil di antara ketiganya.
- AUC hampir mendekati 1.00 di semua kelas (0.99–1.00).
- Ini artinya model sangat mampu membedakan semua kelas dengan sangat baik.
- Keacakan tambahan di Extra Trees membuat pohon-pohon menjadi lebih bervariasi, sehingga overfitting lebih terkontrol tapi tetap presisi tinggi.



[LEARN MORE](#)

Perbandingan

Model	Jenis	Teknik	Class 0	Class 1	Class 2	Rata-rata AUC	Catatan
Decision Tree	Base	Single Tree	1	0.92	0.92	0.95	Masih ada bias antar kelas
Random Forest	Bagging	100 pohon (bootstrap)	1	0.97	0.98	0.98	Lebih seimbang antar kelas
Extra Trees	Bagging	100 pohon (split acak)	1	0.99	0.99	0.99	Paling stabil dan presisi tinggi



Evaluasi Model Learning Curve

Decision Tree

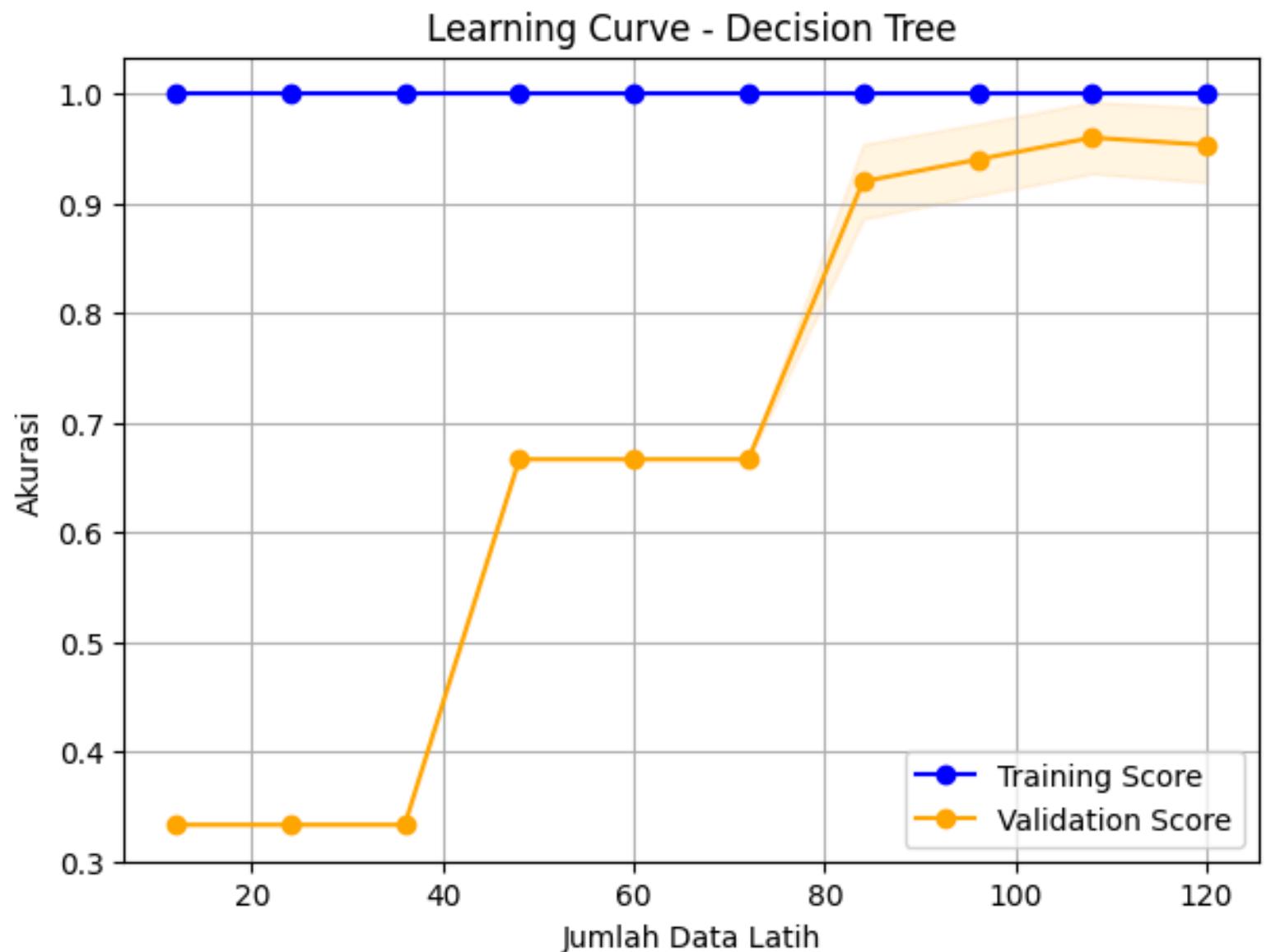
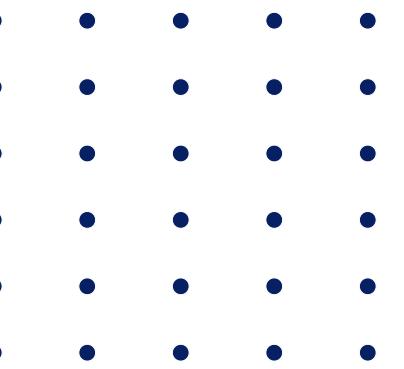
Garis biru (training score): Selalu mendekati 1.0 (100%) sejak jumlah data latih masih sedikit.

- Artinya, Decision Tree mampu menghafal (fit sempurna) data latih bahkan dari subset kecil → tanda overfitting.

Garis oranye (validation score):

- Awalnya sangat rendah (~0.3–0.4), lalu meningkat tajam seiring bertambahnya data.
- Akhirnya mendekati 0.93–0.95, tapi masih di bawah training score.

Gap (jarak) antara kedua kurva cukup besar di awal → tinggi variansi.



[LEARN MORE](#)

Random Forest

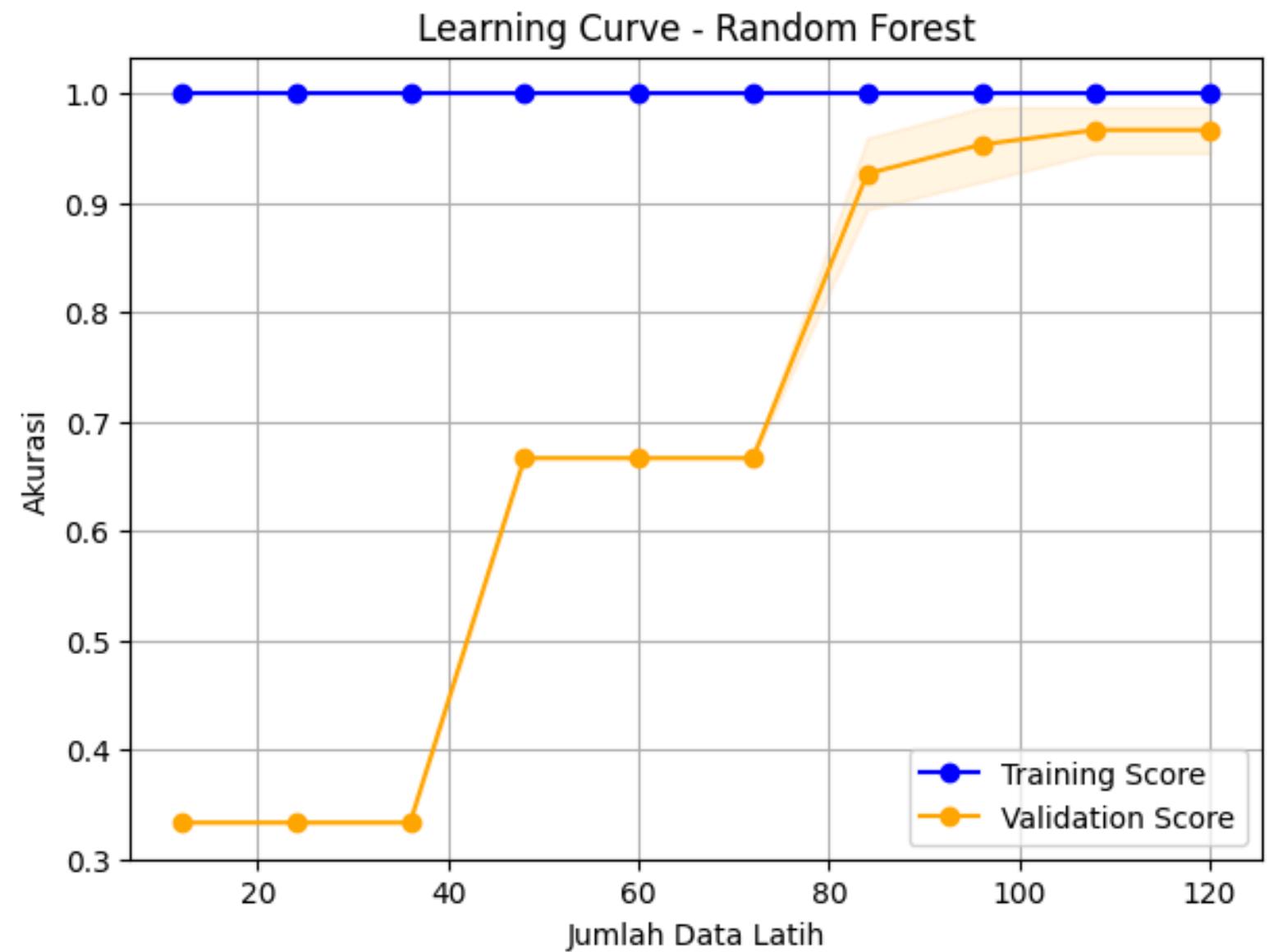
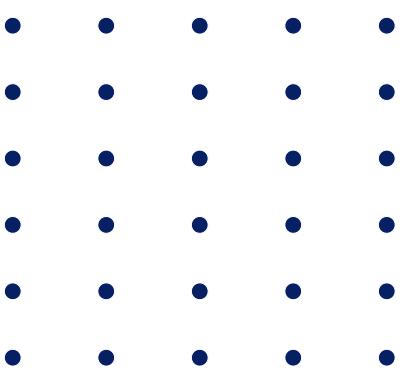
Garis biru (training score): Tetap tinggi (~1.0) tapi sedikit lebih “halus” daripada Decision Tree.

- → Menandakan tiap pohon tidak selalu fit sempurna karena data dilatih secara acak (bootstrap).

Garis oranye (validation score):

- Pola mirip: awalnya rendah (~0.3), naik signifikan saat jumlah data meningkat.
- Akhirnya stabil di sekitar 0.95–0.97 dengan deviasi kecil.

Gap antara training dan validation lebih kecil dibanding Decision Tree → model lebih stabil.



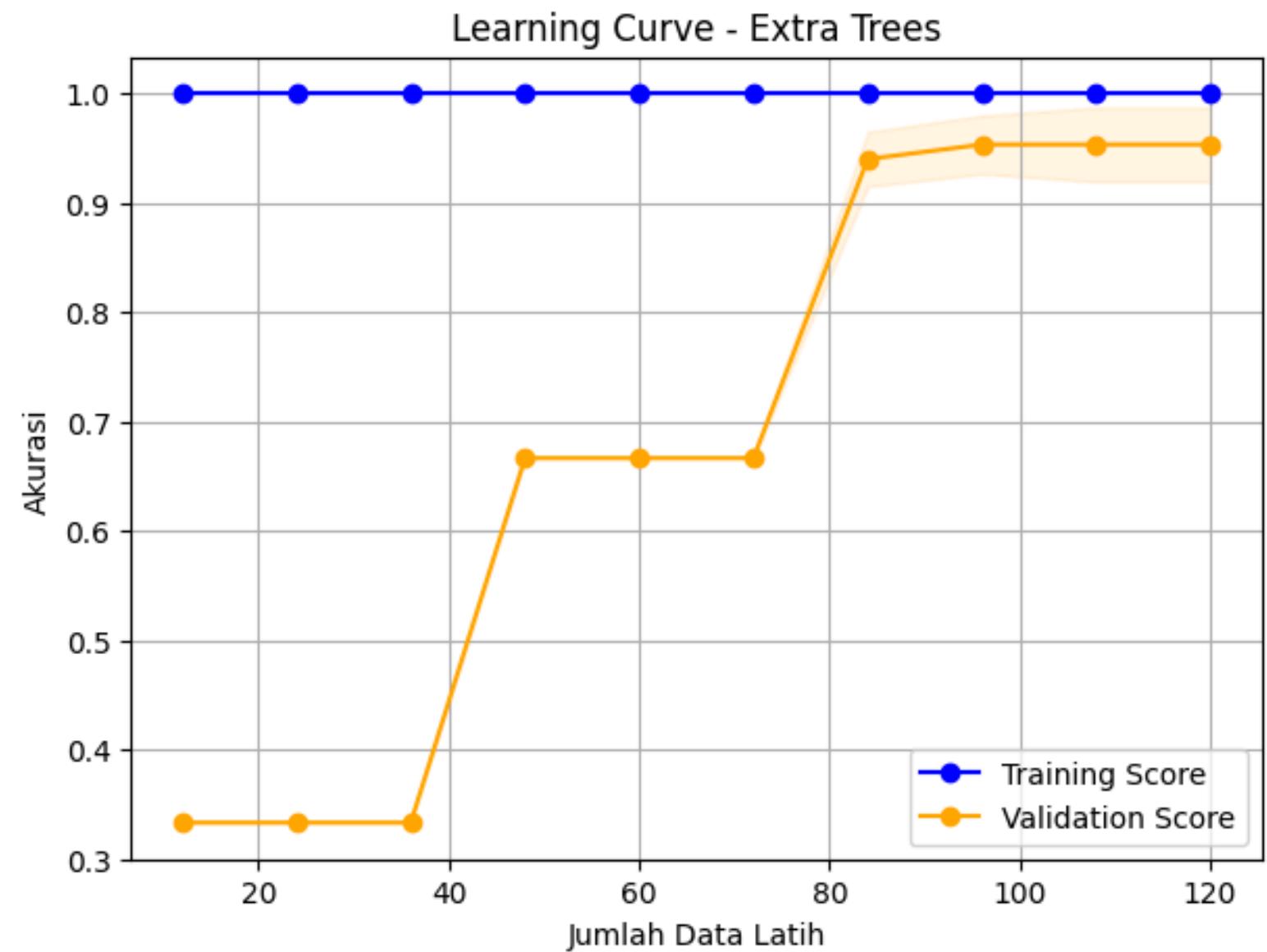
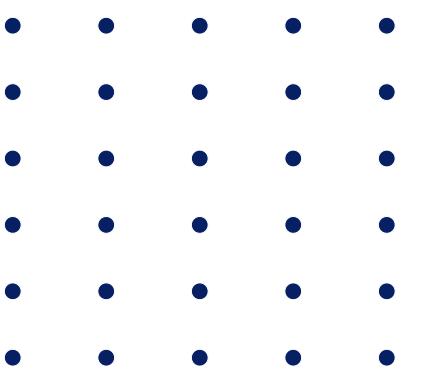
[LEARN MORE](#)

Extra Trees

Kurva hampir identik dengan Random Forest, tapi:

- Validation score sedikit lebih stabil (fluktuasi lebih kecil).
- Deviasi (area bayangan) paling sempit → variansi paling rendah.

Menandakan Extra Trees menghasilkan model yang paling cepat konvergen dan paling stabil di antara ketiganya.



[LEARN MORE](#)

Perbandingan

Model	Jenis	Pola Training	Pola Validation	Gap	Kesimpulan
Decision Tree	Base	1.0 (flat, overfit)	Naik lambat ke 0.93	Lebar	Overfitting, variansi tinggi
Random Forest	Bagging	Stabil di 1.0	Naik cepat ke 0.95	Lebih kecil	Variansi menurun, lebih stabil
Extra Trees	Bagging (lebih acak)	Stabil di 1.0	Naik cepat ke 0.96	Paling kecil	Variansi rendah, generalisasi terbaik

Data Heart Attack

Melakukan analisis pada dataset Heart Attack dengan:
Membandingkan base classifier (Decision Tree) dengan dua model bagging classifier
(Random Forest & Extra Trees), lalu menganalisis performa mereka menggunakan
Confusion Matrix, ROC/AUC, atau Learning Curve.

Confusion
Matrix

ROC/AUC

Learning Curve

Preprocessing

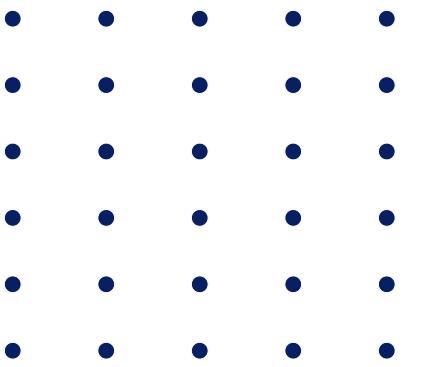
```
[13] 0s
▶ # Pisahkan fitur dan target
X = df.drop(columns=['output'])
y = df['output']

# Standarisasi fitur
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split data
X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, test_size=0.2, random_state=42, stratify=y
)
|
print("\nUkuran Data:")
print(f"X_train: {X_train.shape}, X_test: {X_test.shape}")

...
Ukuran Data:
X_train: (242, 13), X_test: (61, 13)
```

Aspek	Keterangan
Jumlah total data	303 sampel (242 train, 61 test)
Jumlah fitur (X)	13 atribut medis
Jenis masalah	Klasifikasi biner (0 = sehat, 1 = berisiko)
Tujuan preprocessing	Menyeimbangkan skala fitur & memisahkan data untuk training dan evaluasi
Hasil	Dataset sudah siap untuk modeling dengan Decision Tree, Random Forest, dan Extra Trees



LEARN MORE

inisialisasi Model

• • • •
• • • •
• • • •
• • • •
• • • •

```
# === 6. Inisialisasi Model ===
models = {
    'Decision Tree': DecisionTreeClassifier(random_state=42),
    'Random Forest': RandomForestClassifier(n_estimators=100, random_state=42),
    'Extra Trees': ExtraTreesClassifier(n_estimators=100, random_state=42)
}
```

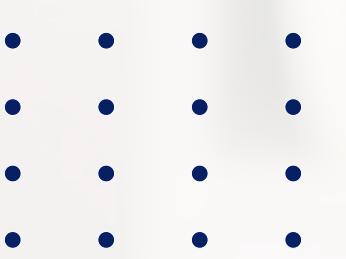
Baris kode ini adalah inti dari perbandingan model bagging:

- Decision Tree → baseline model tunggal.
- Random Forest → Bagging klasik (menggabungkan banyak pohon).
- Extra Trees → Bagging dengan tingkat randomisasi lebih tinggi.

LEARN MORE

www.reallygreatsite.com

Evaluasi Model Matrix Confusi



Decision Tree

Model: Decision Tree					
	precision	recall	f1-score	support	
0	0.71	0.61	0.65	28	
1	0.70	0.79	0.74	33	
accuracy			0.70	61	
macro avg	0.71	0.70	0.70	61	
weighted avg	0.71	0.70	0.70	61	

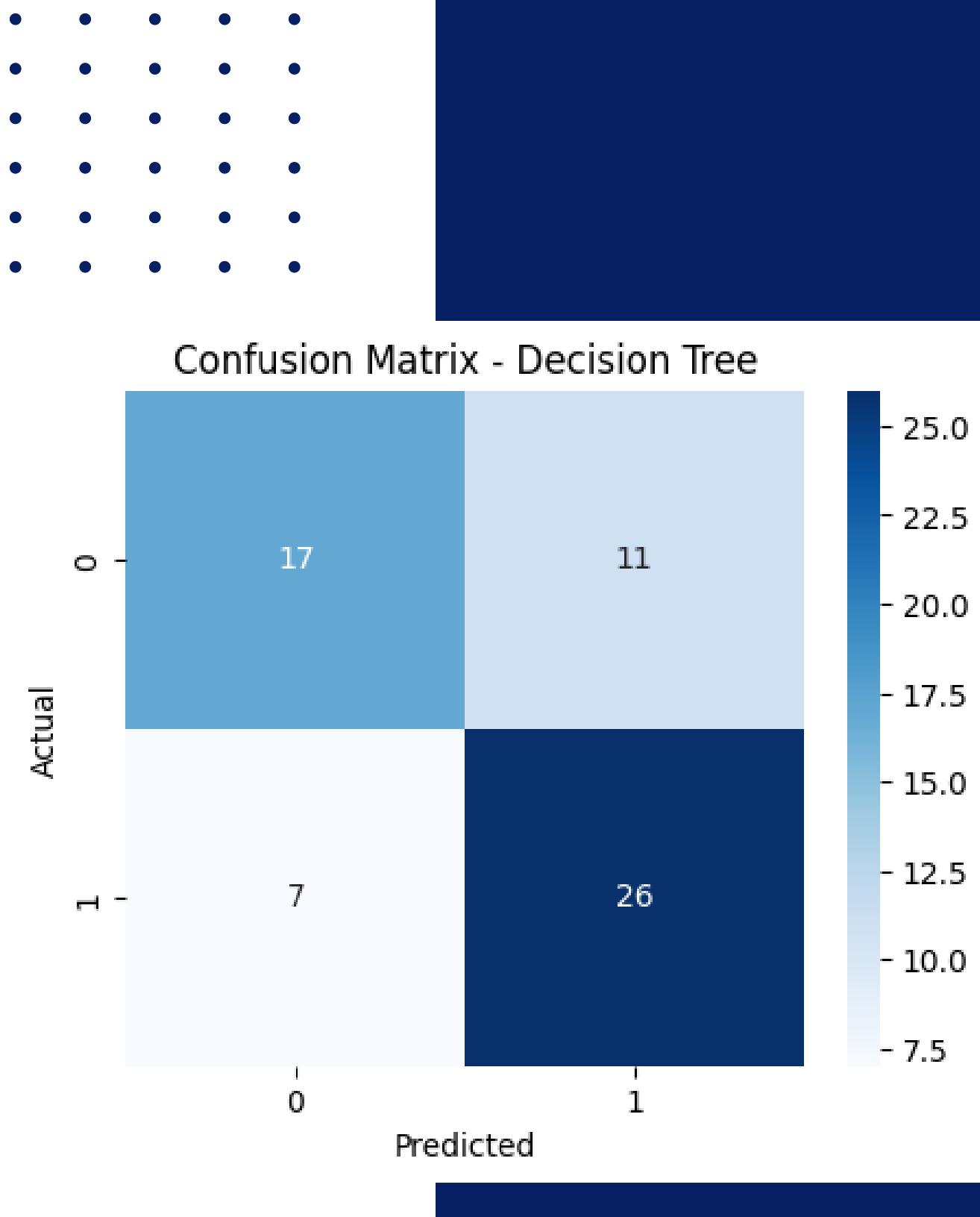
Akurasi = 70%

Precision (0.71) → Dari semua pasien yang diprediksi “sakit”, 71% benar-benar sakit.

Recall (0.79 untuk kelas 1) → Model berhasil mendeteksi 79% pasien yang benar-benar sakit.

Kesalahan utama:

- 11 pasien sehat diprediksi salah sebagai sakit (False Positive).
- 7 pasien sakit gagal dideteksi (False Negative).

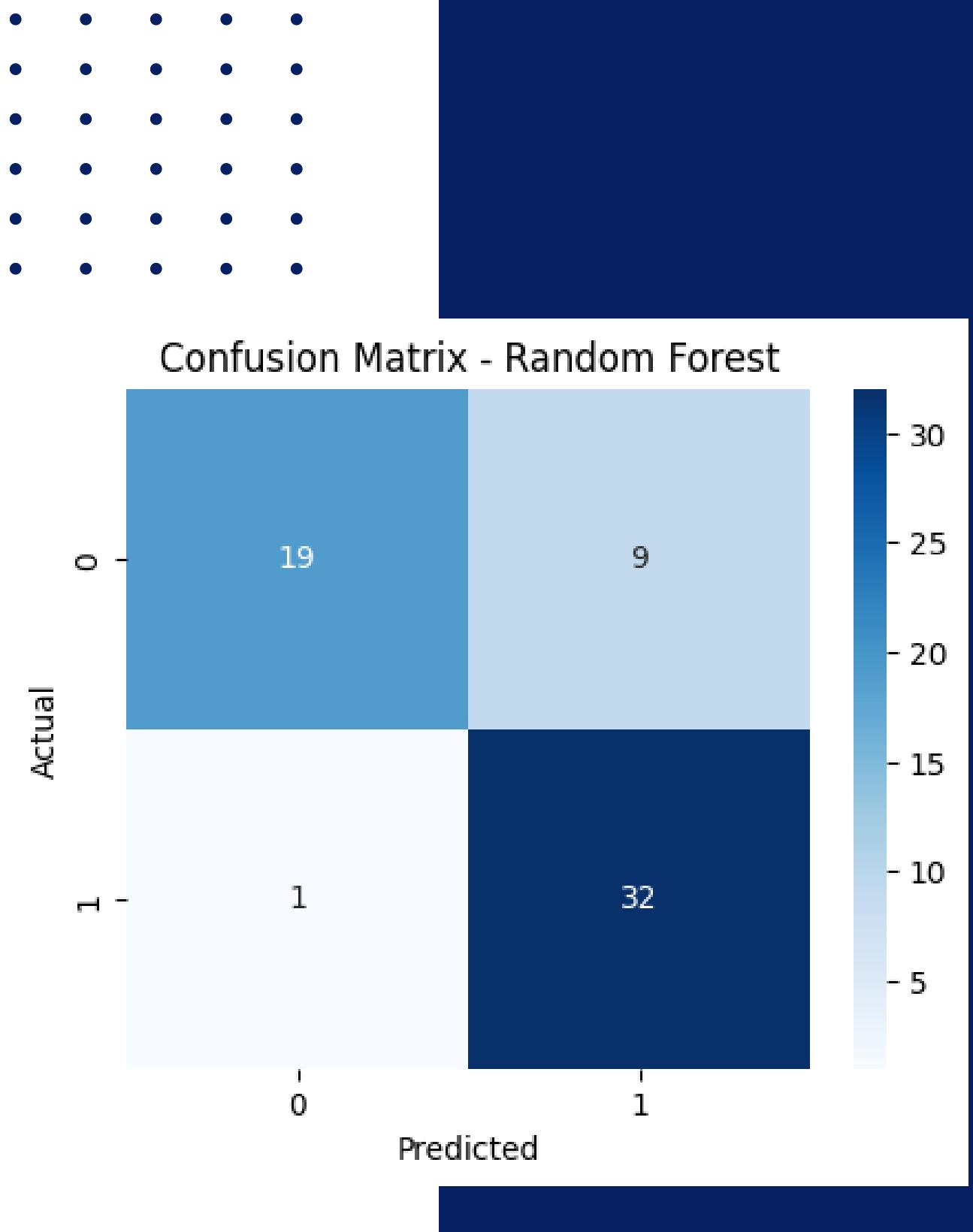


[LEARN MORE](#)

Random Forest

Model: Random Forest				
	precision	recall	f1-score	support
0	0.95	0.68	0.79	28
1	0.78	0.97	0.86	33
accuracy			0.84	61
macro avg	0.87	0.82	0.83	61
weighted avg	0.86	0.84	0.83	61

- Akurasi = 84%
- Precision (1: 0.78) → 78% prediksi “sakit” benar-benar sakit.
- Recall (1: 0.97) → model sangat baik mengenali pasien sakit (hampir semua terdeteksi).
- False Negative (FN) hanya 1 → sangat bagus untuk kasus medis, karena pasien sakit hampir tidak ada yang lolos deteksi.

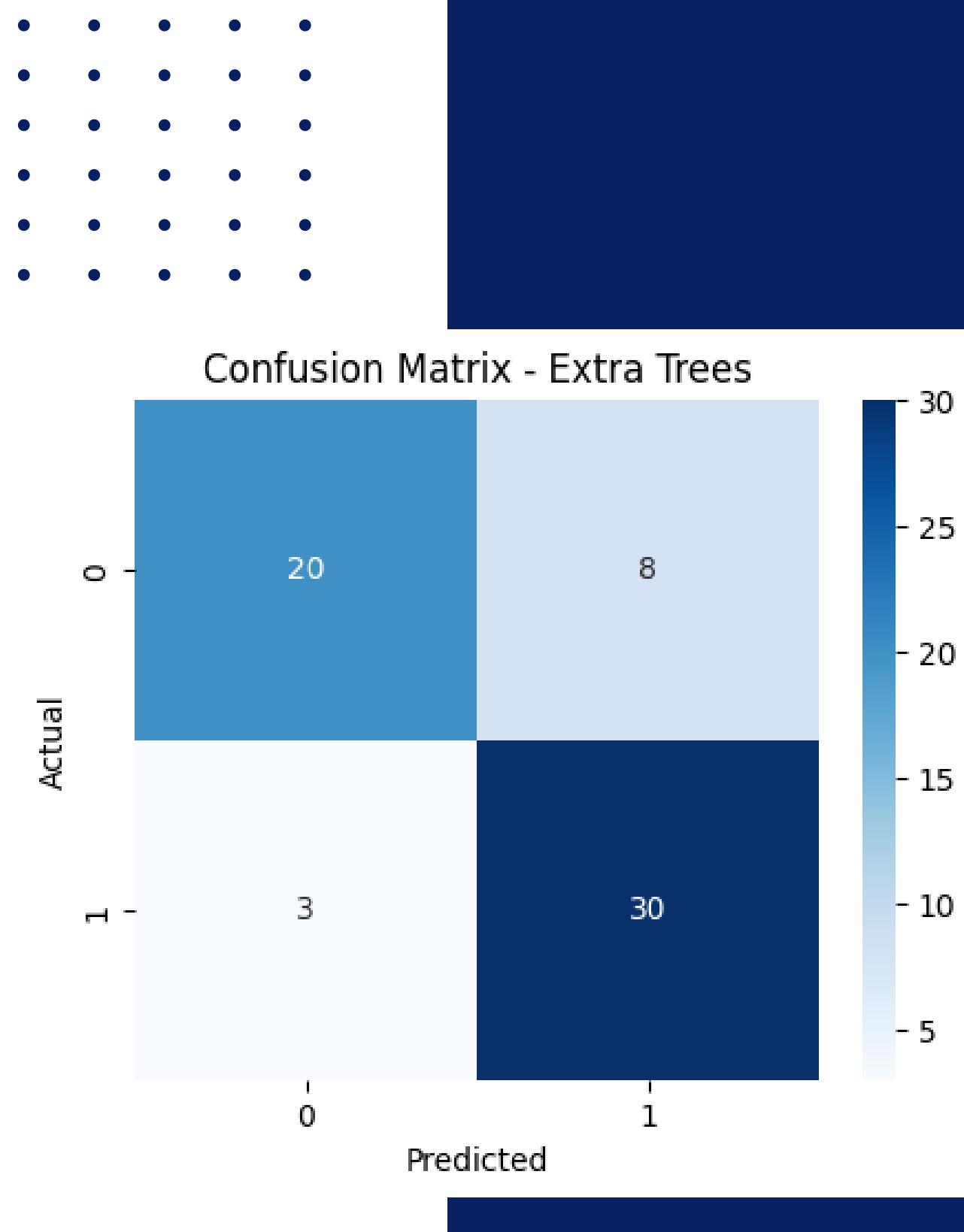


LEARN MORE

Extra Trees

Model: Extra Trees				
	precision	recall	f1-score	support
0	0.87	0.71	0.78	28
1	0.79	0.91	0.85	33
accuracy			0.82	61
macro avg	0.83	0.81	0.81	61
weighted avg	0.83	0.82	0.82	61

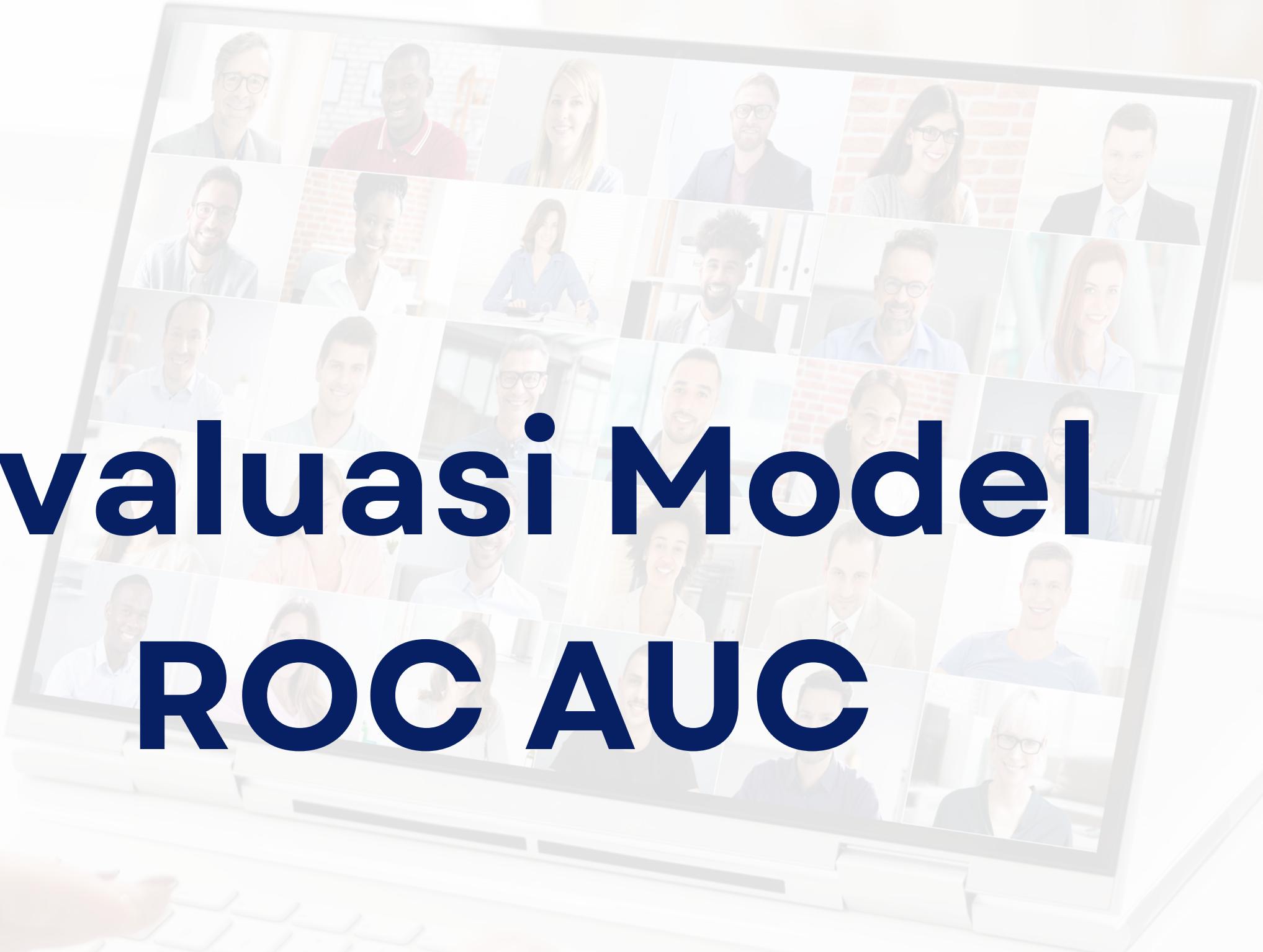
- Akurasi = 82%
- Precision (1: 0.79) → mirip dengan Random Forest.
- Recall (1: 0.91) → sedikit menurun dari Random Forest, tapi masih sangat baik.
- FN meningkat sedikit (dari 1 ke 3), namun FP menurun (dari 9 ke 8).



LEARN MORE

Perbandingan

Model	Akurasi	Precision (1)	Recall (1)	F1-Score (1)	Catatan
Decision Tree	0.7	0.7	0.79	0.74	Dasar, cenderung overfit
Random Forest	0.84	0.78	0.97	0.86	Paling akurat, recall tinggi
Extra Trees	0.82	0.79	0.91	0.85	Stabil, cepat, seimbang



Evaluasi Model ROC AUC

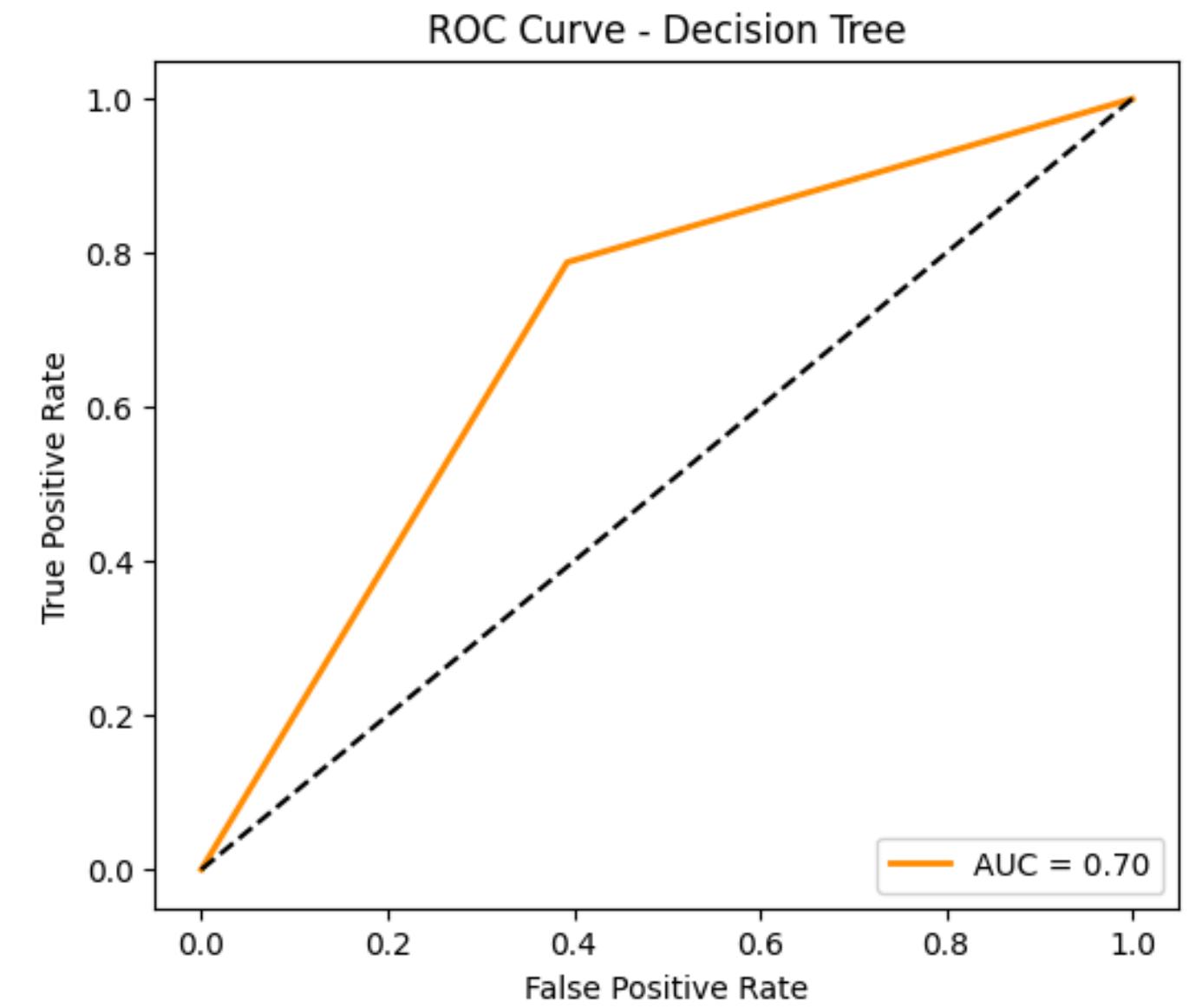
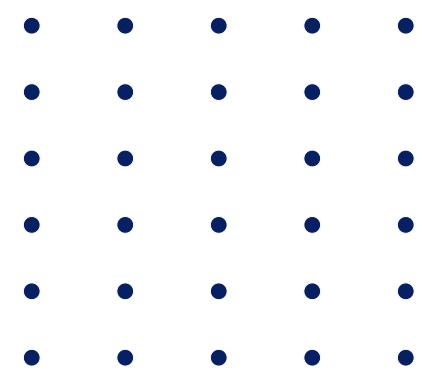
Decision Tree

ROC Curve:

- Garisnya cukup tajam tapi belum tinggi → hanya sedikit di atas garis diagonal (model acak).
- AUC = 0.70 → artinya kemampuan model membedakan pasien sakit dan sehat sekitar 70% benar.

Interpretasi:

- Decision Tree masih kesulitan membedakan dua kelas dengan baik.
- Masih terlalu overfit – hasil di training bagus, tapi di testing performanya turun.
- Model ini belum stabil: bisa berubah drastis jika data sedikit diubah.



[LEARN MORE](#)

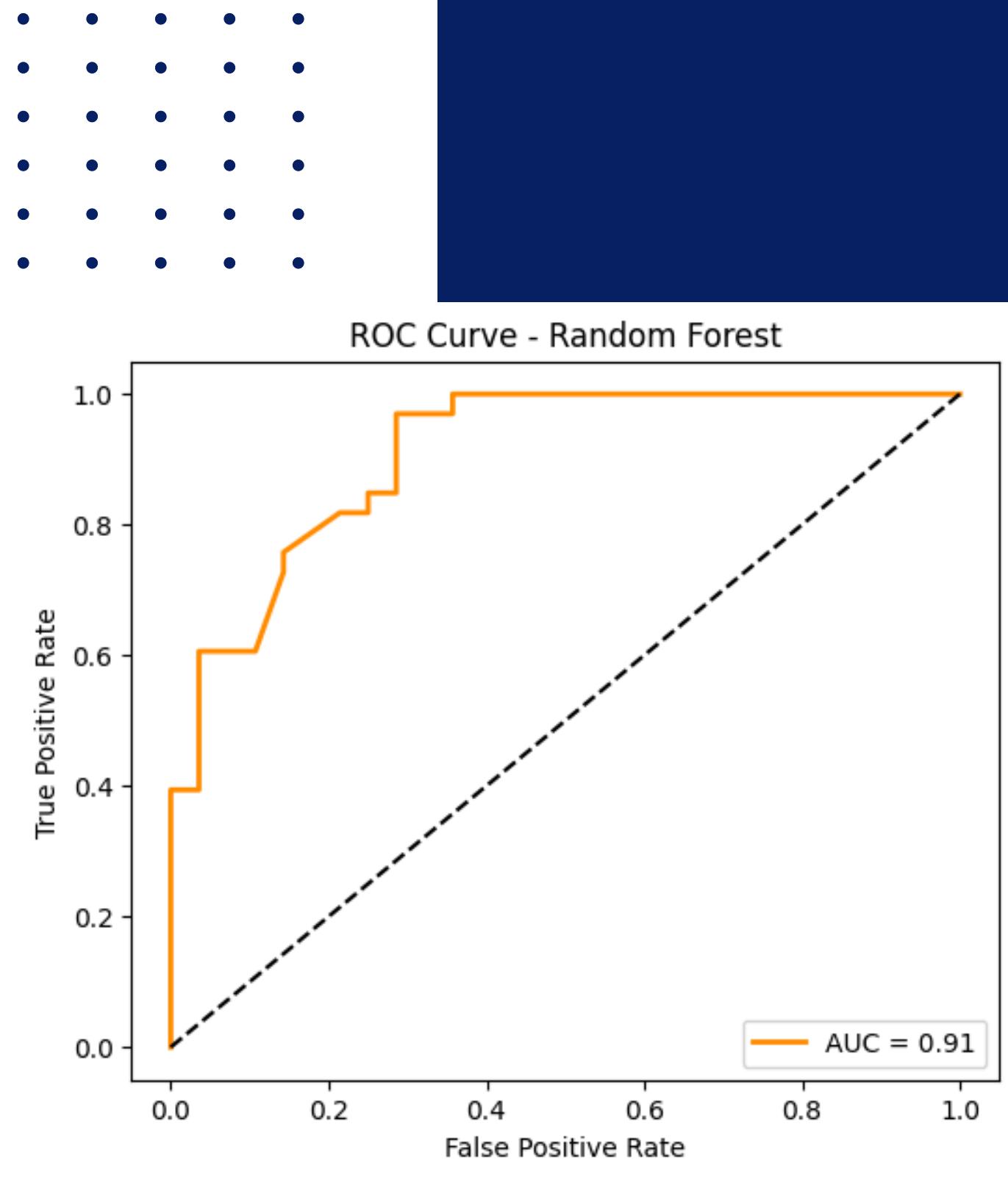
Random Forest

ROC Curve:

- Kurva mendekati sudut kiri atas → artinya model mampu menjaga FPR rendah sambil mempertahankan TPR tinggi.
- AUC = 0.91 → sangat tinggi, menandakan kemampuan klasifikasi excellent.

Interpretasi:

- Random Forest sangat efektif dalam membedakan pasien sakit dan sehat.
- Dengan bagging dan voting antar banyak pohon:
- Model mengurangi noise,
- Menyeimbangkan bias-variance tradeoff,
- Dan mencegah overfitting yang sering terjadi pada pohon tunggal.



[LEARN MORE](#)

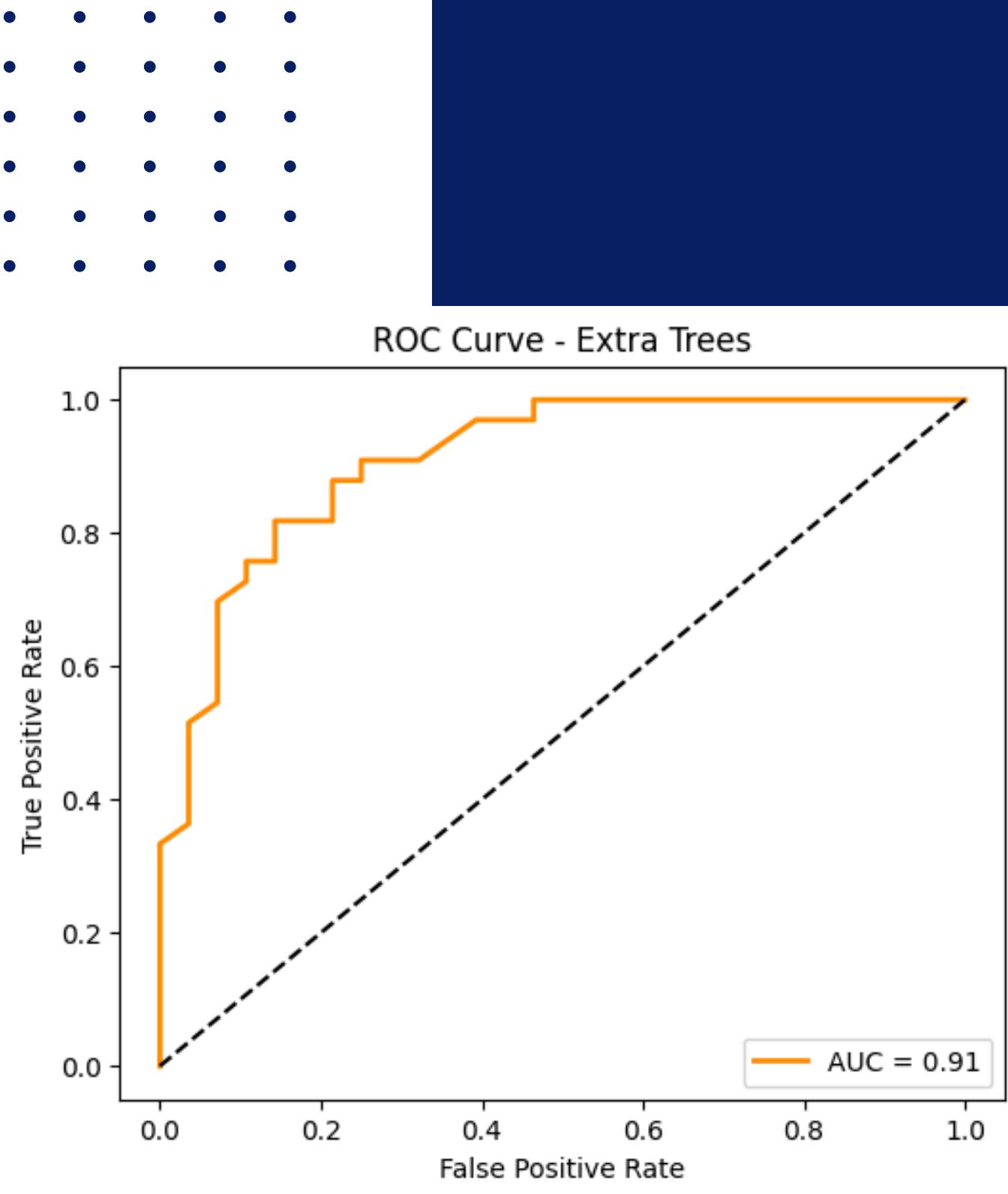
Extra Trees

ROC Curve:

- Hampir identik dengan Random Forest.
- Kurva juga mendekati kiri atas.
- AUC = 0.91, menunjukkan performa sebanding.

Interpretasi:

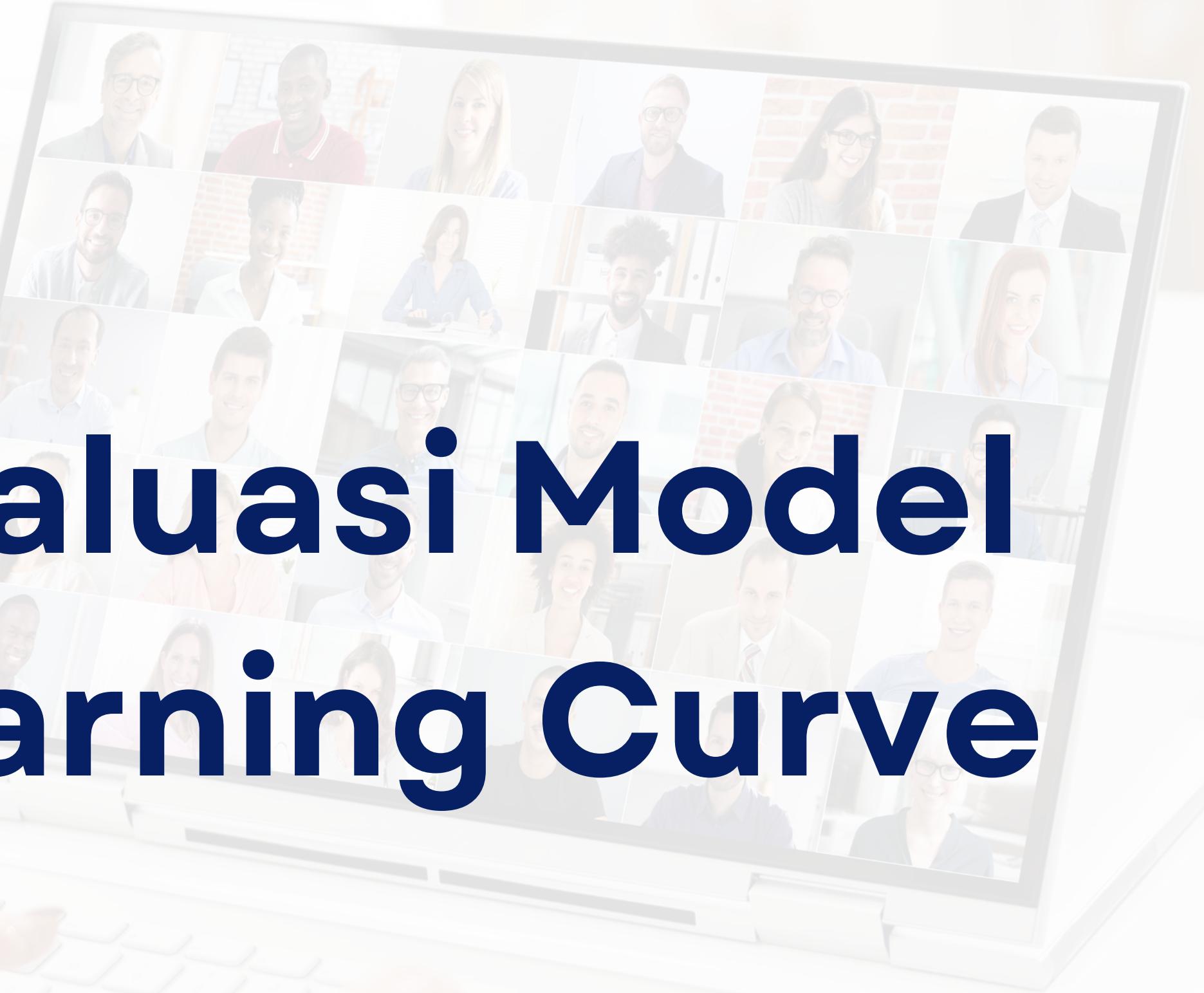
- Extra Trees melakukan lebih banyak randomisasi pada titik split, tapi hasilnya tetap sangat baik.
- Model ini lebih cepat dilatih daripada Random Forest, karena tidak mencari titik split terbaik setiap kali.
- Secara generalisasi, Extra Trees lebih stabil dan efisien, walau kadang AUC-nya bisa sedikit fluktuatif tergantung dataset.



[LEARN MORE](#)

Perbandingan

Model	AUC Score	Kemampuan Membedakan Kelas	Interpretasi
Decision Tree	0.7	Sedang	Masih sulit membedakan pasien sakit & sehat
Random Forest	0.91	Sangat Baik	Akurat, stabil, dan minim kesalahan
Extra Trees	0.91	Sangat Baik	Setara dengan Random Forest, tapi lebih efisien



Evaluasi Model Learning Curve

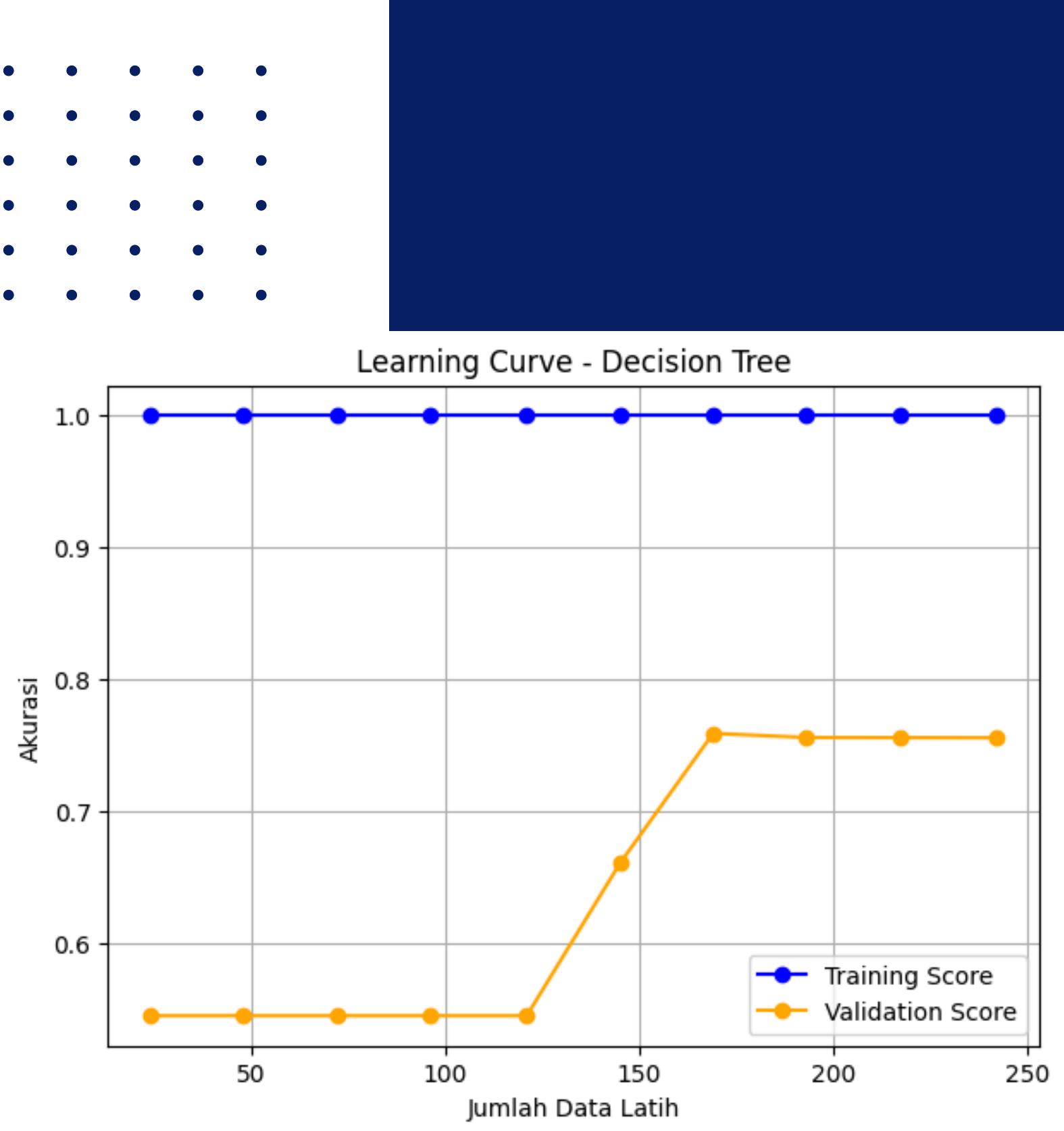
Decision Tree

Pola Grafik:

- Training Score: konstan di 1.0 (100%) untuk semua ukuran data → model menghafal data training.
- Validation Score: mulai dari ~0.55 dan naik sedikit hingga 0.75.

Analisis:

- Overfitting yang kuat.
 - Model terlalu fokus pada data latih, tapi gagal menggeneralisasi ke data baru.
 - Terlihat dari gap besar antara training (1.0) dan validation (~0.75).
- Penambahan data tidak banyak memperbaiki kinerja.
- Hal ini wajar karena Decision Tree tunggal sangat sensitif terhadap noise dan variasi data.



[LEARN MORE](#)

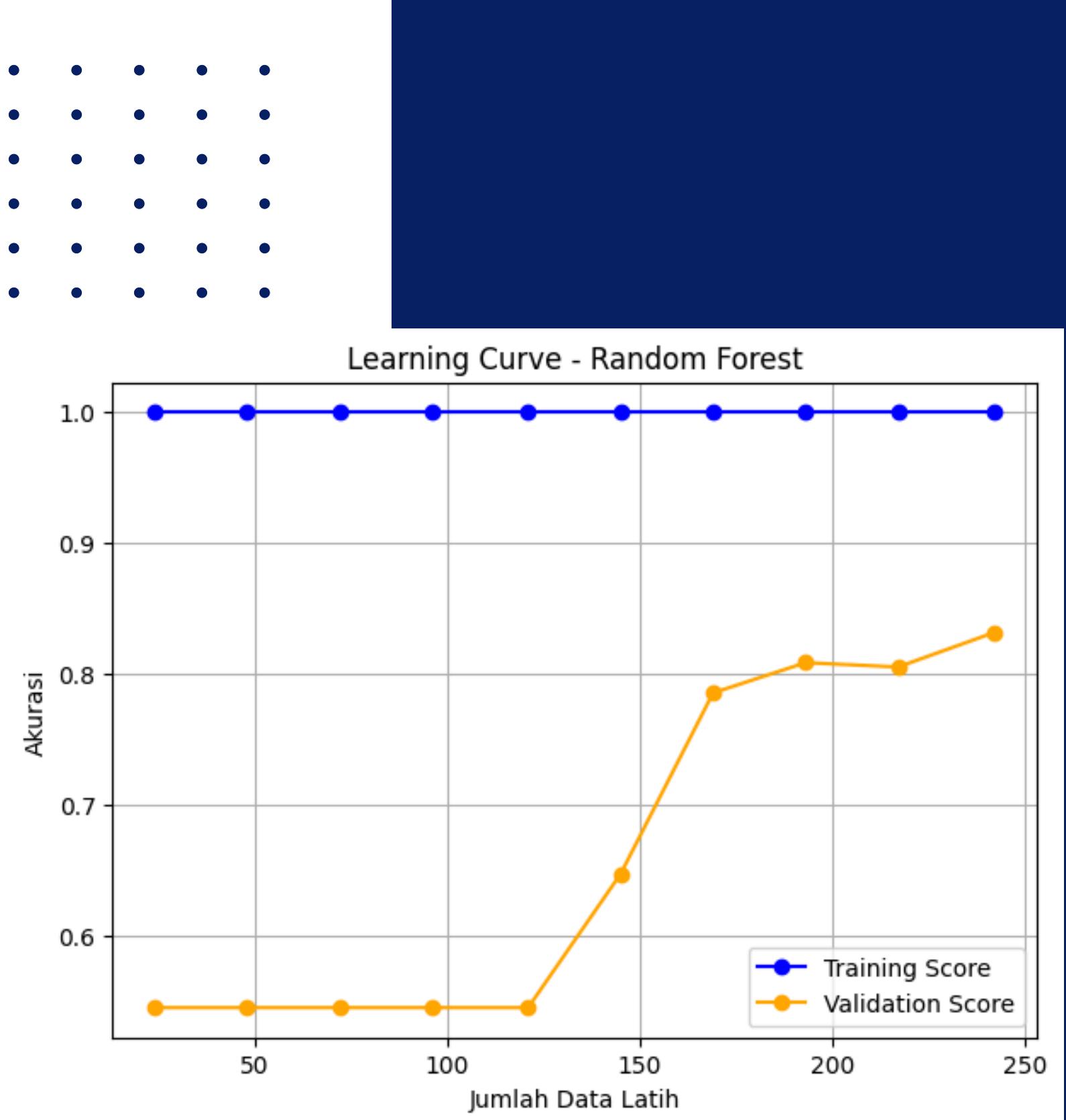
Random Forest

Pola Grafik:

- Training Score: tetap sangat tinggi (~1.0) tapi sedikit lebih stabil.
- Validation Score: meningkat konsisten dari 0.55 → sekitar 0.82 ketika jumlah data makin besar.
- Jarak antara training dan validation lebih kecil dibanding Decision Tree.

Analisis:

- Random Forest berhasil mengurangi overfitting melalui bagging (gabungan banyak pohon acak).
- Validation score meningkat dan mulai stabil → model generalisasi jauh lebih baik.
- Garis training yang hampir datar di atas dan validation yang naik perlahan menunjukkan model stabil dan kuat terhadap data tambahan.



[LEARN MORE](#)

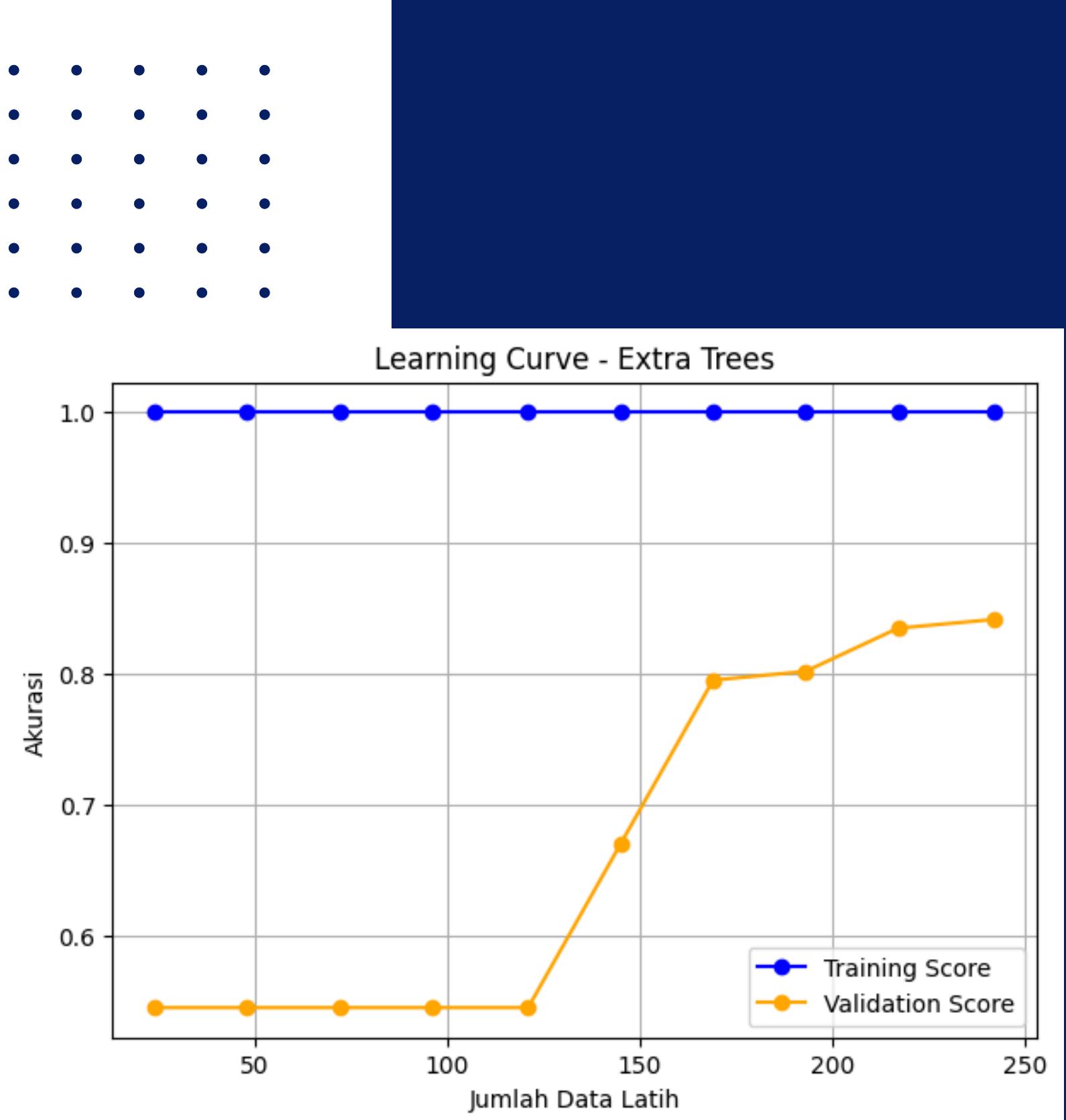
Extra Trees

Pola Grafik:

- Pola hampir identik dengan Random Forest.
- Training Score: konstan di 1.0.
- Validation Score: naik dari 0.55 hingga 0.83–0.85, sedikit lebih tinggi dari Random Forest di titik akhir.

Analisis:

- Extra Trees juga bagging model, tapi dengan split point yang dipilih secara acak penuh.
- Akibatnya, model lebih cepat dilatih dan cenderung lebih general.
- Jarak antara training dan validation bahkan sedikit lebih kecil daripada Random Forest → menandakan overfitting paling kecil di antara ketiganya.



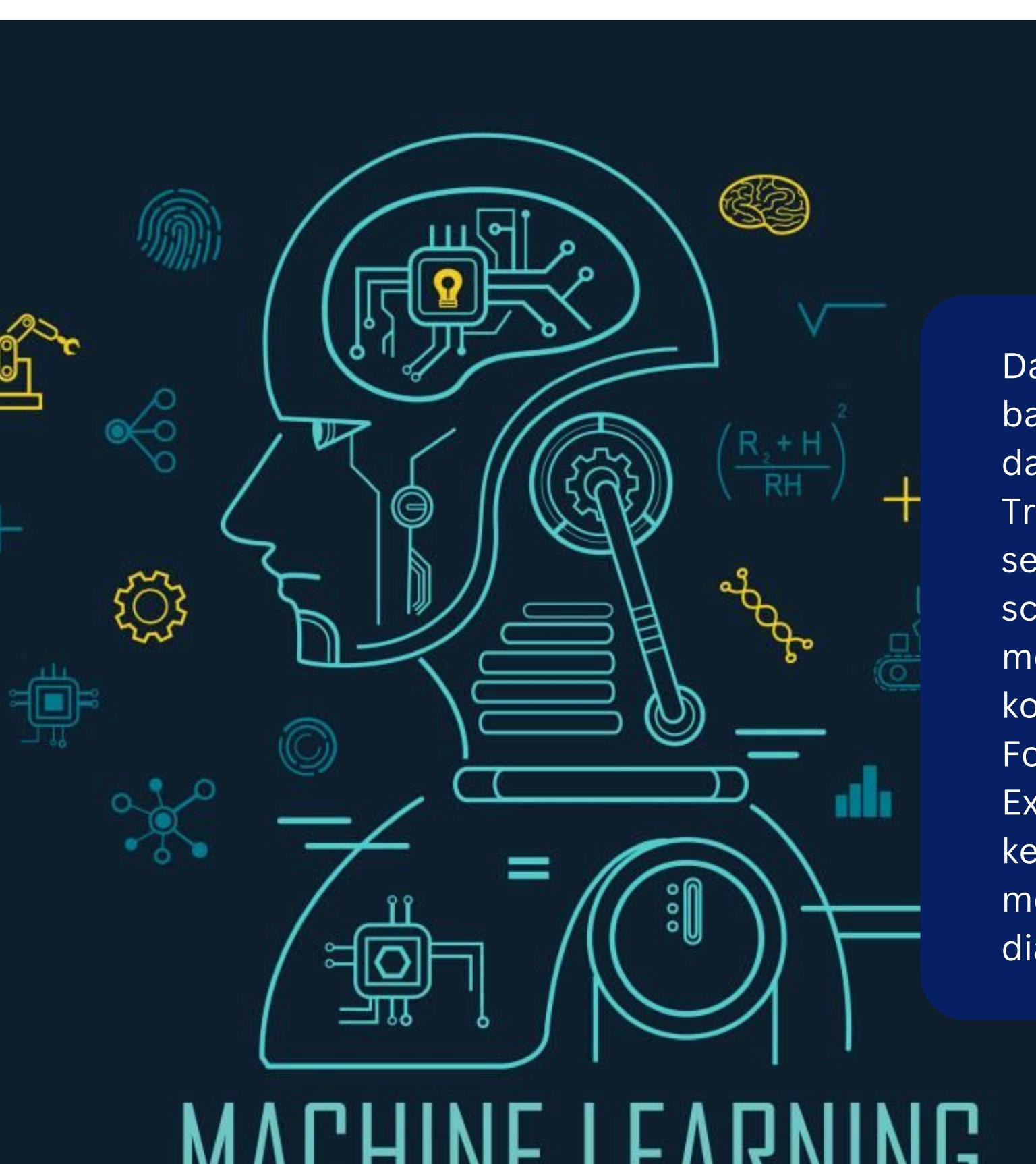
[LEARN MORE](#)

Perbandingan

Model	Training Score	Validation Score	Pola	Diagnosa
Decision Tree	1	~0.75	Gap besar	Overfitting berat
Random Forest	1	~0.82	Gap mengecil	Generalisasi baik
Extra Trees	1	~0.83–0.85	Gap kecil	Generalisasi sangat baik

Conclusion

Dari seluruh hasil analisis terhadap dataset Heart Attack, dapat disimpulkan bahwa metode ensemble learning berbasis bagging seperti Random Forest dan Extra Trees memberikan performa jauh lebih baik dibandingkan Decision Tree tunggal. Decision Tree cenderung overfitting, dengan akurasi training sempurna namun kemampuan generalisasi rendah (AUC 0.70 dan validation score sekitar 0.75). Sebaliknya, Random Forest dan Extra Trees mampu mengurangi overfitting dan memberikan hasil yang lebih stabil, akurat, serta konsisten, dengan AUC tinggi (0.91) dan akurasi validasi di atas 0.80. Random Forest unggul dalam mendekripsi pasien berisiko (recall tinggi), sedangkan Extra Trees menawarkan efisiensi dan stabilitas yang baik. Secara keseluruhan, penggunaan bagging classifier terbukti efektif dalam meningkatkan kinerja model dan membuat prediksi lebih andal untuk kasus diagnosis penyakit jantung.



Thank You For Your Attention



Dimensionality Reduction

Unsupervised Learning

Classification

Supervised Learning

Machine Learning

Reinforcement Learning



Source Code

Data Iris : [Klik Disini](#)

Data Heart Attack : [Klik Disini](#)