M. Fikri Avishena Parinduri

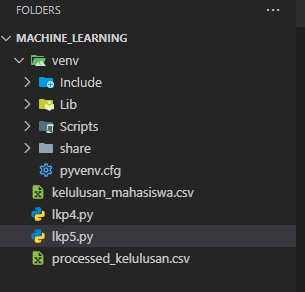
231011401029

05TPLE016

Machine Learning

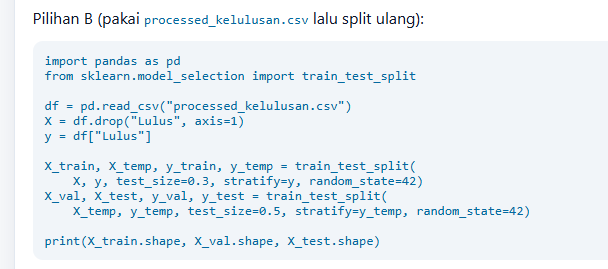
Lembar Kerja Pertemuan 5

Disini saya membuat file baru yaitu lkp5.py

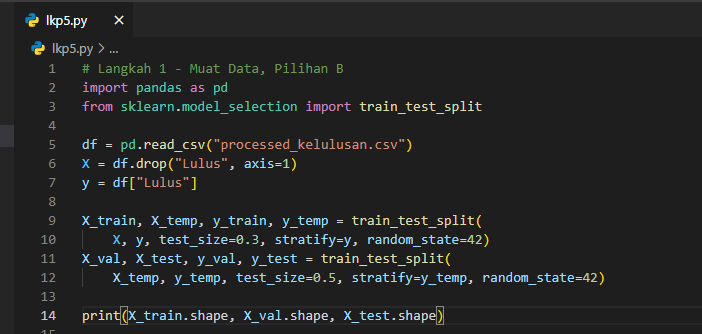


1. Langkah 1 – Muat Data

Saya pakai Pilihan B (pakai processed\_kelulusan.csv lalu split ulang):



Kode:



Output:



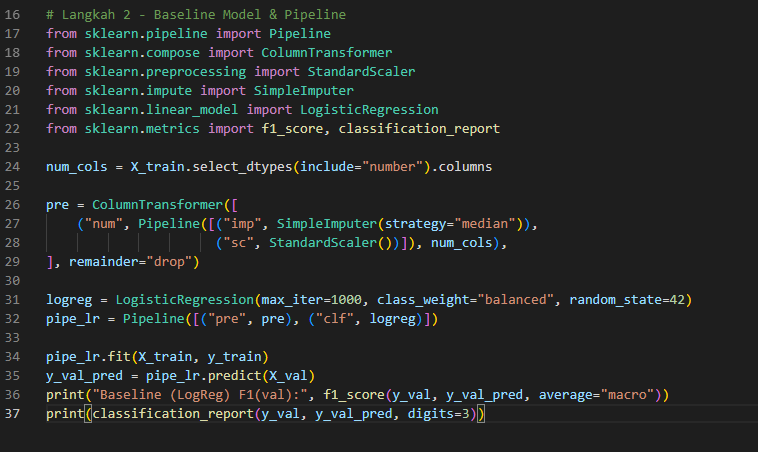
Penjelasan:

* Data dibaca dari file hasil pra-proses (processed\_kelulusan.csv)
* Target: kolom Lulus
* Split menjadi:
  + 70% → train
  + 15% → validation
  + 15% → test
* Menggunakan stratify → memastikan distribusi kelas tetap seimbang.

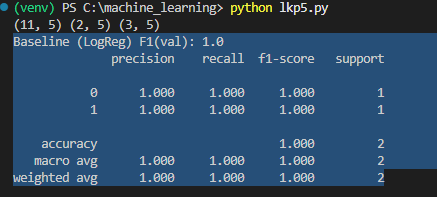
Aman selama jumlah data tiap kelas ≥ 2 di setiap subset, jadi dataset sudah cukup (≥14–16 baris).

1. Langkah 2 – Baseline Model & Pipeline

Kode nya:



Output:

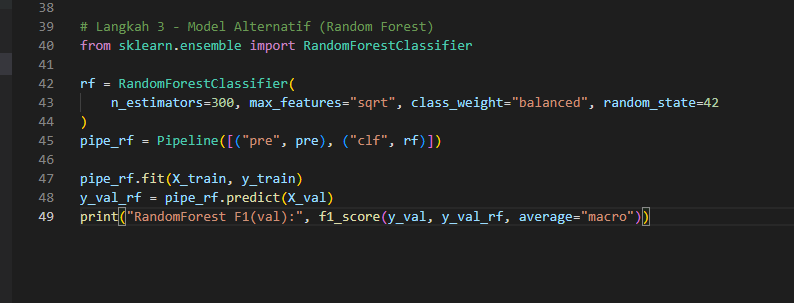


Menggunakan Pipeline dan ColumnTransformer:

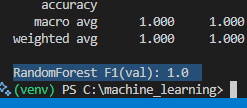
* SimpleImputer(strategy="median") → menangani nilai kosong
* StandardScaler() → menstandarkan skala numerik
* LogisticRegression(class\_weight="balanced") → model awal, menangani ketidakseimbangan kelas

1. Langkah 3 – Model Alternatif (Random Forest)

Kode nya:



Output:



Penjelasan:

1. Model dibuat:
   * RandomForestClassifier dengan 300 pohon keputusan.
   * max\_features="sqrt" berarti setiap pohon hanya melihat sebagian fitur saat membentuk split (meningkatkan variasi antar pohon).
   * class\_weight="balanced" membuat model menyesuaikan bobot kelas agar seimbang — penting kalau data tidak seimbang antara Lulus=1 dan Lulus=0.
   * random\_state=42 menjaga hasil tetap konsisten.
2. Pipeline digunakan:
   * pre: pipeline preprocessing yang berisi imputasi (SimpleImputer) dan standardisasi (StandardScaler).
   * clf: model Random Forest itu sendiri.
   * Jadi, ketika pipe\_rf.fit() dijalankan, data akan otomatis diimputasi dan diskalakan sebelum dilatih.
3. Prediksi & evaluasi:
   * Model dipakai untuk memprediksi data validasi (y\_val\_rf).
   * Skor F1 dihitung menggunakan average="macro" agar rata-rata F1 antar kelas dihitung secara seimbang.

Mengapa hasilnya F1(val): 1.0?

Artinya model memprediksi semua data validasi dengan benar 100% — tidak ada kesalahan klasifikasi.

Namun, dalam konteks dataset kecil seperti punyamu (hanya 10 data asli, bahkan mungkin <10 di train/val/test setelah split), hal ini sangat mungkin terjadi karena overfitting.

Pembahasan:

* 1. Ukuran dataset sangat kecil

Dari 10 baris data:

* 70% train = 7 data
* 15% val = 1.5 → dibulatkan jadi 1 atau 2 data
* 15% test = 1 atau 2 data

Dengan jumlah data sekecil itu, model Random Forest bisa dengan mudah menghafal semua pola, terutama karena setiap fitur (IPK, Jumlah\_Absensi, Waktu\_Belajar\_Jam, dll.) memiliki hubungan kuat dan mungkin tanpa noise.

* 1. Pola data terlalu “bersih”

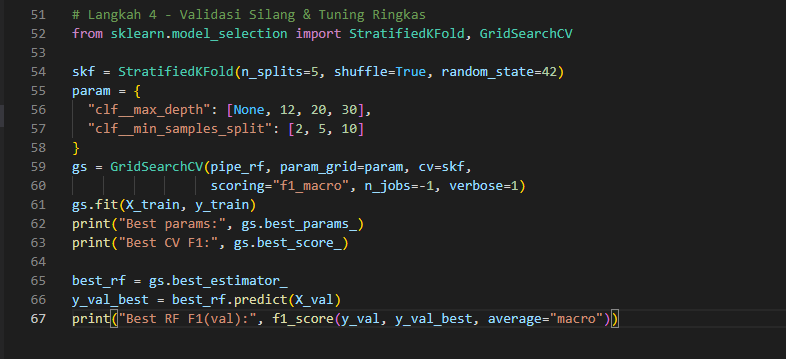
Dari CSV awal, terlihat bahwa mahasiswa dengan:

* IPK tinggi dan waktu belajar panjang → Lulus = 1
* IPK rendah dan absensi tinggi (sering absen) → Lulus = 0

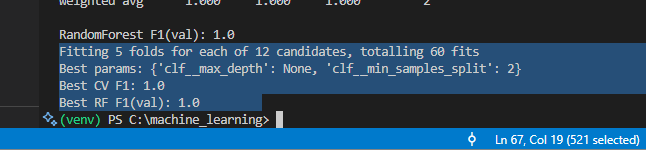
Hubungan ini nyaris linier sempurna. Random Forest sangat baik menangkap pola deterministik seperti ini, jadi ia bisa memisahkan kelas dengan sempurna.

1. Langkah 4 – Validasi Silang & Tuning Ringkas

Kode nya:



Output:

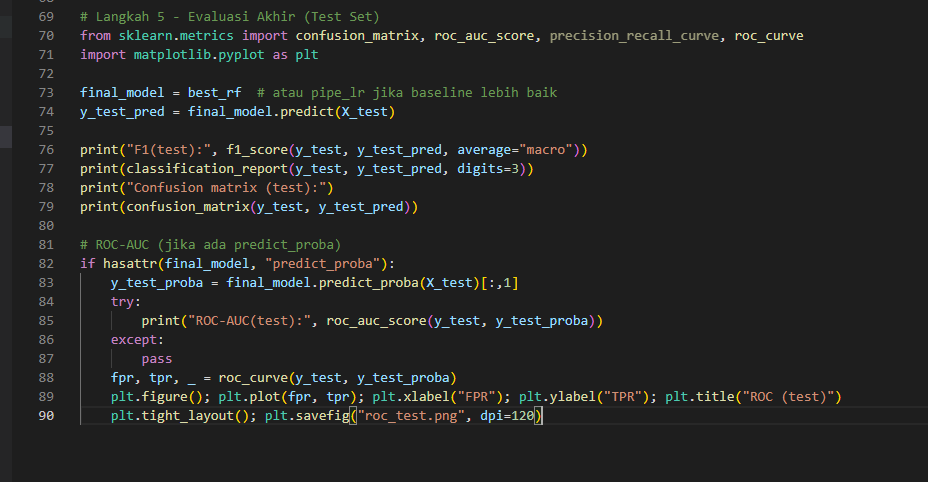


Penjelasan:

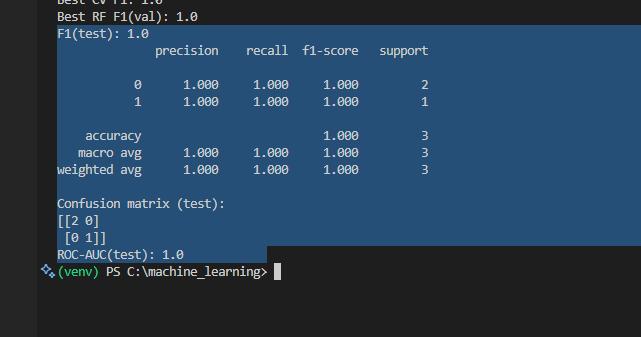
* GridSearchCV menguji berbagai kombinasi max\_depth dan min\_samples\_split
* StratifiedKFold(5) memastikan tiap lipatan memiliki distribusi kelas seimbang
* scoring="f1\_macro" → menilai performa rata-rata antar kelas

1. Langakh 5 – Evaluasi Akhir (Test Set)

Kode nya:



Output:



Penjelasan:

Import Matriks evaluasi

* confusion\_matrix → menunjukkan jumlah prediksi benar/salah per kelas.
* roc\_auc\_score → menilai kemampuan model membedakan antara kelas (semakin mendekati 1, semakin baik).
* precision\_recall\_curve dan roc\_curve → dipakai untuk menggambar kurva performa model.
* matplotlib.pyplot → digunakan untuk membuat grafik ROC.

Pilih model final

* + Model terbaik hasil tuning (best\_rf) dari langkah sebelumnya.
  + Tapi juga bisa ganti ke pipe\_lr (Logistic Regression baseline) jika performanya lebih stabil.

Prediksi pada test set

* + Model digunakan untuk memprediksi kelas (Lulus atau tidak) pada data yang belum pernah dilihat sama sekali selama pelatihan.
  + Tujuan: menguji generalisasi model, bukan kemampuannya menghafal data.

Hitung metrik performa utama

* + F1 Score
  + Kombinasi dari precision dan recall.
  + Dengan average="macro", nilai F1 tiap kelas dihitung lalu dirata-ratakan sama rata (tanpa melihat jumlah datanya).
  + Classification Report

precision: Ketepatan: dari semua prediksi positif, berapa yang benar.

recall: Keberhasilan tangkap: dari semua data positif, berapa yang berhasil ditemukan model.

f1-score: Rata-rata harmonik antara precision & recall.

support: Jumlah sampel pada kelas itu di data test.

* + Confusion Matrix

Artinya:

* TN (True Negative) → Prediksi 0, aktual 0
* FP (False Positive) → Prediksi 1, tapi salah
* FN (False Negative) → Prediksi 0, tapi seharusnya 1
* TP (True Positive) → Prediksi 1, aktual 1

Dengan matriks ini bisa tahu kesalahan spesifik model, bukan hanya skor rata-rata.

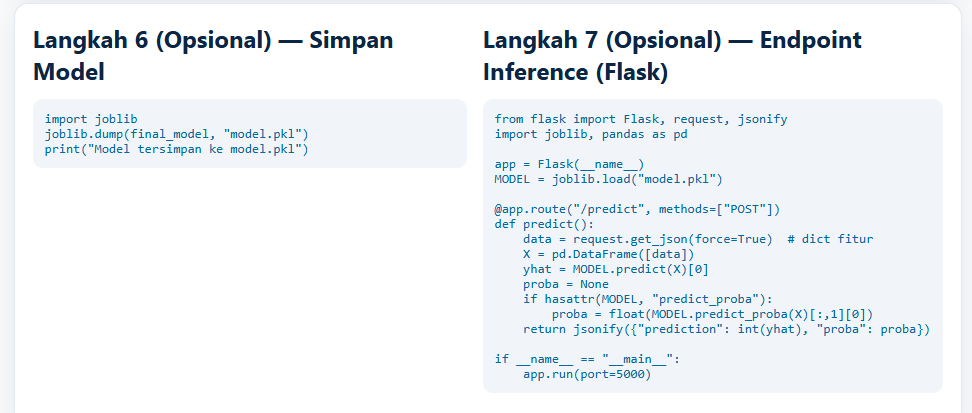
ROC dan AUC

* + **predict\_proba**
* Random Forest dan Logistic Regression sama-sama punya metode ini.
* Menghasilkan probabilitas prediksi, bukan hanya label (misal: 0.92 artinya 92% yakin kelas “Lulus”).
  + **ROC-AUC**
* ROC (Receiver Operating Characteristic) mengukur trade-off antara:
  + **TPR (True Positive Rate)** = recall
  + **FPR (False Positive Rate)** = kesalahan prediksi positif pada kelas negatif
* **AUC (Area Under Curve)** menunjukkan seberapa baik model membedakan dua kelas:
  + 0.5 → tidak lebih baik dari tebak acak
  + 1.0 → sempurna

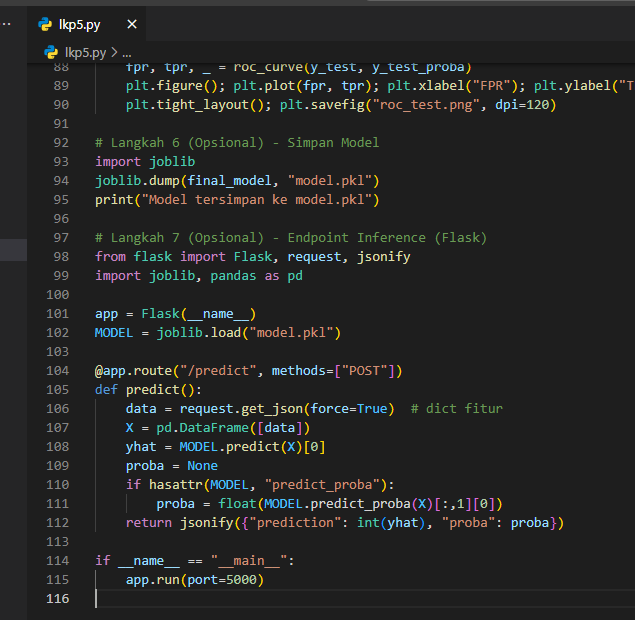
Plot ROC Curve

* Kurva ini membantu memvisualisasikan performa model dalam berbagai ambang batas keputusan.
* Disimpan ke file roc\_test.png agar bisa kamu lihat hasilnya nanti.

Lalu ada Langkah opsional:

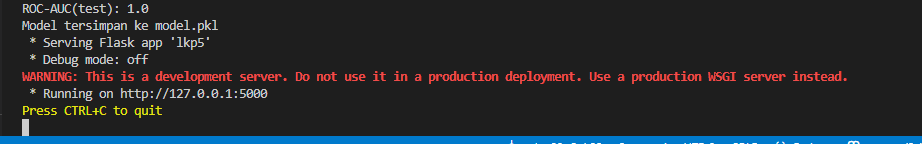


Masukan kode Langkah 6 dan 7 tersebut



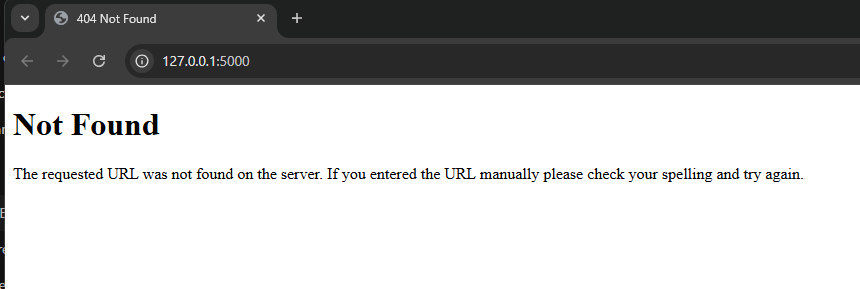
Di Langkah 6 akan menghasilkan file “model.pkl” berisi pipeline lengkap (preprocessing + model).  
Aman dan siap digunakan untuk inference.

Hasilnya untuk di Langkah 6 dan 7:



File berhasil disimpan

Terlihat di terminal berhasil jalan web local



Web nya di buka jadi seperti ini