M. Fikri Avishena Parinduri

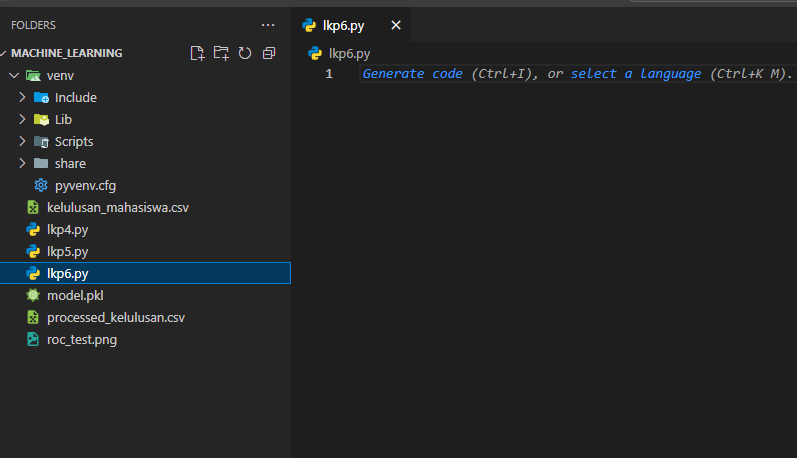
231011401029

05TPLE016

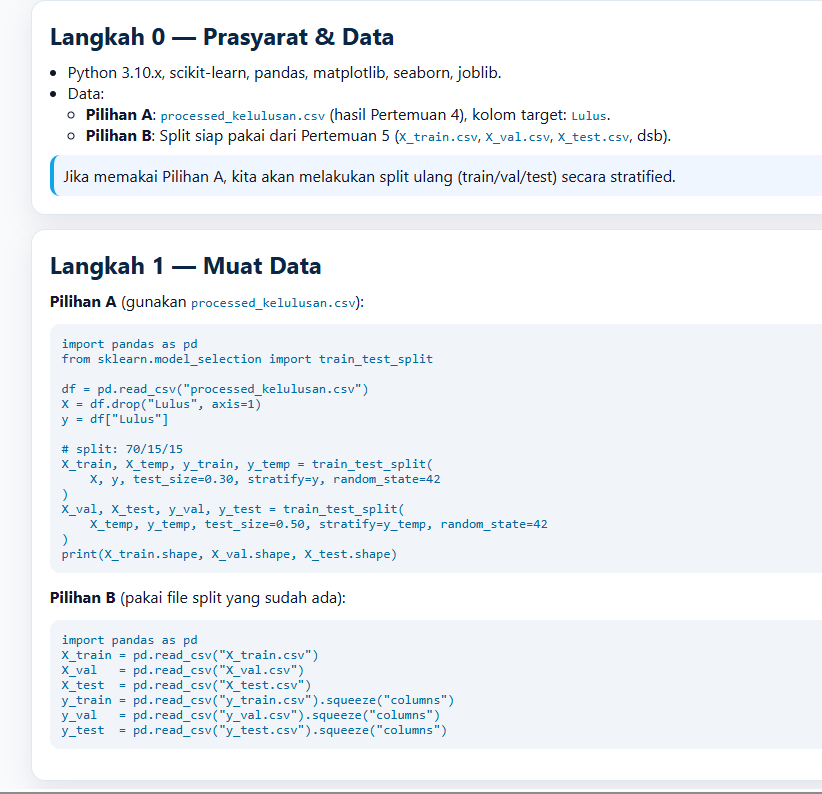
Machine Learning

Lembar Kerja Pertemuan 6

Disini saya membuat file baru yaitu lkp6.py

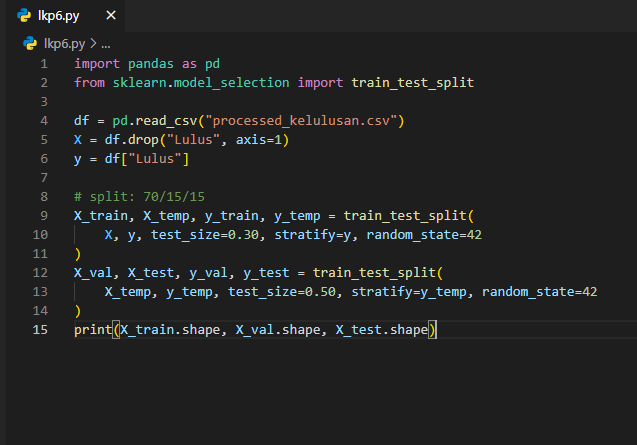


Saya akan menjalankan Pilihan A di Langkah 0 dan itu pasti ke Pilihan A di Langkah 1

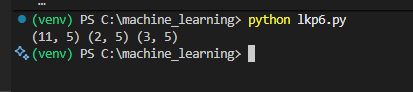


1. Langkah 1 – Muat Data

Kode nya:



Outputnya:



Penjelasan:

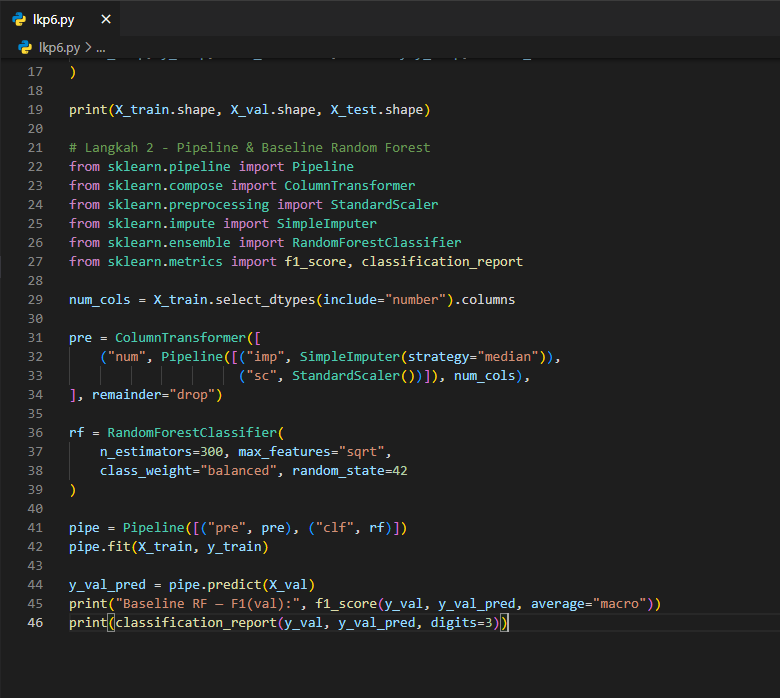
* pandas digunakan untuk membaca file CSV ke dalam DataFrame.
* Dataset diasumsikan sudah diproses (processed\_kelulusan.csv) sehingga siap dipakai untuk model.
* Lulus adalah kolom target (label) — biasanya 0 atau 1 (tidak lulus / lulus).
* X berisi semua fitur (variabel independen), sedangkan y berisi label keluaran.

Lalu di split

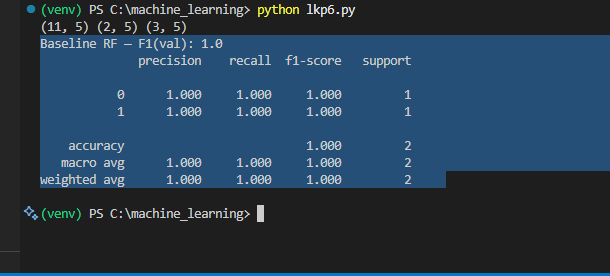
* Dataset dibagi **70% train**, **15% validation**, dan **15% test**.
* stratify=y memastikan proporsi kelas (misal lulus/tidak lulus) tetap seimbang di setiap subset.
* random\_state=42 membuat hasil pembagian tetap sama setiap dijalankan.
* Output shape menunjukkan ukuran data di setiap subset.

1. Langkah 2 – Pipeline & Baseline

Kode nya:



Output:



Berhasil

Penjelasan:

* SimpleImputer(strategy="median"): mengisi nilai kosong dengan median tiap kolom.
* StandardScaler(): menormalkan data (mean = 0, std = 1) agar model lebih stabil.
* ColumnTransformer: memastikan hanya kolom numerik yang diproses, kolom lain dibuang.

Lanjut ke bagian RandomForest

Pejnelasan:

* RandomForestClassifier: model ensemble berbasis banyak decision tree.
* n\_estimators=300: jumlah pohon yang digunakan.
* class\_weight="balanced": menyesuaikan bobot kelas otomatis jika data tidak seimbang.
* Pipeline: menggabungkan preprocessing (pre) dan model (clf) dalam satu alur otomatis.

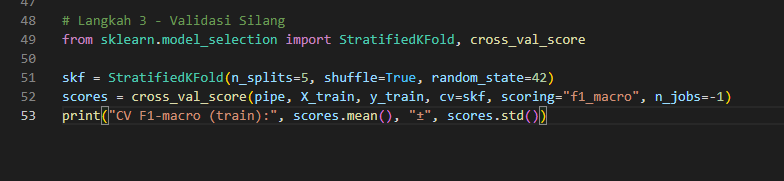
Kemudian evaluasi (baseline) pada validation test

Penjelasan:

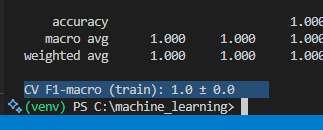
* predict() menghasilkan prediksi label untuk data validasi.
* f1\_score(..., average="macro"): rata-rata F1 dari semua kelas (tanpa memperhatikan proporsi kelas).
* classification\_report: menampilkan precision, recall, dan F1 per kelas.

1. Langkah 3 – Validasi Silang

Kode:



Output:

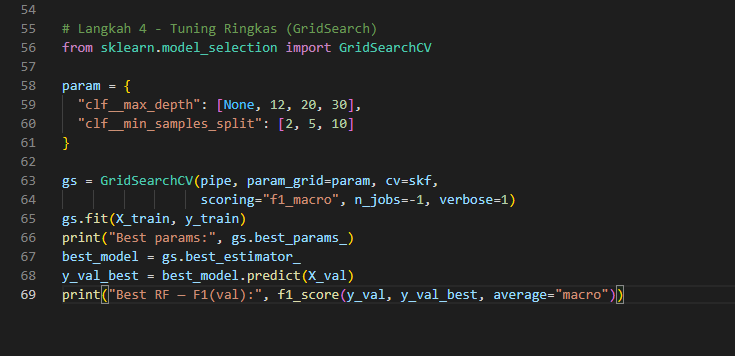


Penjelasan:

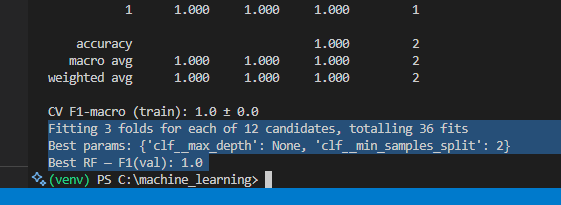
* StratifiedKFold: membagi data training menjadi 3 bagian (fold) dengan proporsi kelas seimbang.
* cross\_val\_score: melatih dan menguji model pada setiap fold.
* Menghasilkan rata-rata skor F1 sebagai ukuran kestabilan performa model di data train.

1. Langkah 4 – Tuning Ringkas (GridSearch)

Kode nya:



Output nya:



Penjelasan:

* GridSearchCV mencoba berbagai kombinasi parameter untuk mencari yang terbaik.
* Parameter yang dicoba di sini:
  + max\_depth: kedalaman maksimum tiap pohon.
  + min\_samples\_split: jumlah minimal sampel untuk memecah node.
* verbose=1: menampilkan progress di konsol.
* Hasil terbaik disimpan di gs.best\_estimator\_.

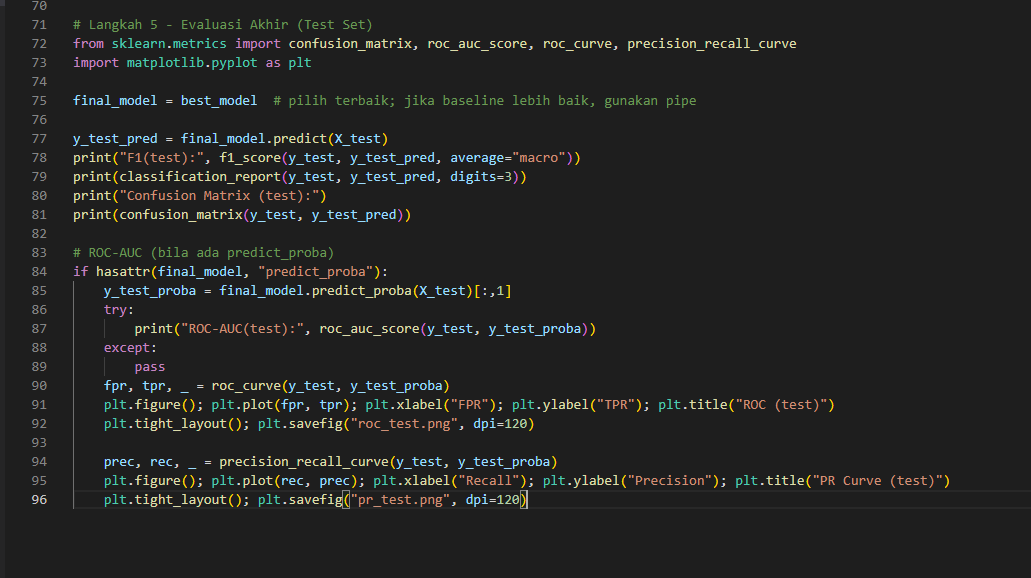
Lalu lanjut ke evaluasi hasil tuning

Penjelasan:

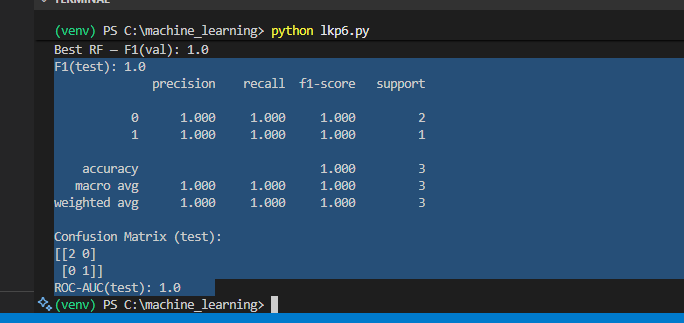
* Menampilkan kombinasi parameter terbaik.
* Mengevaluasi kembali pada validation set dengan model hasil tuning.

1. Langkah 5 – Evaluasi Akhir (Test Set)

Kode nya:



Ouptut nya:

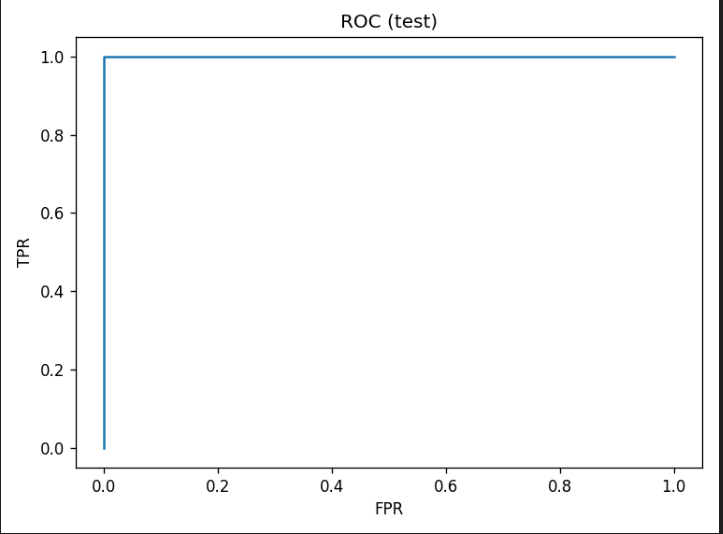


Menghasilkan image juga dengan nama:

roc\_test.png dan pr\_test.png

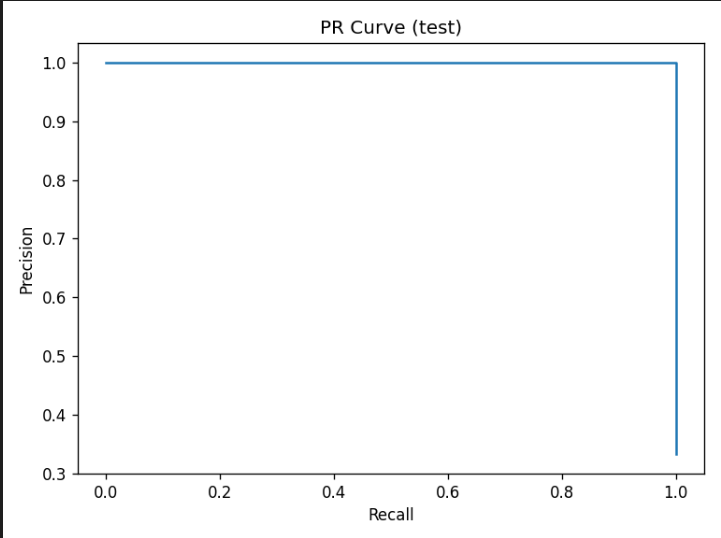
Pertama saya akan buka roc\_test.png

Isinya:



Lalu untuk pr\_test.png

Isinya:



Penjelasan:

* Menguji model final di data test yang belum pernah dilihat model sebelumnya.
* Mengukur performa akhir (umumnya untuk laporan).
* confusion\_matrix: menunjukkan jumlah benar/salah prediksi tiap kelas.

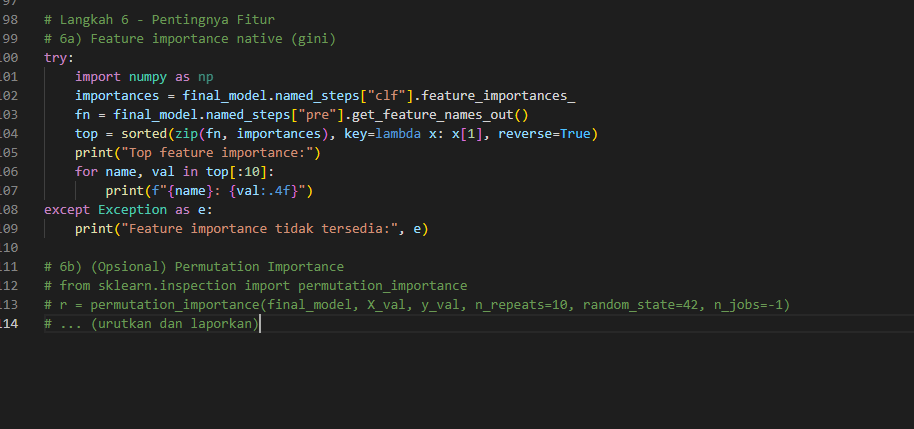
Dilanjut ke ROC & Precision-Recall Curve

Penjelasan:

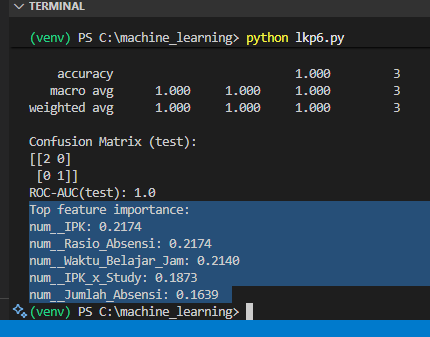
* predict\_proba: memberi probabilitas keyakinan model, bukan sekadar label 0/1.
* Digunakan untuk:
  + roc\_auc\_score: area di bawah kurva ROC (semakin tinggi semakin baik).
  + roc\_curve dan precision\_recall\_curve: menggambar kurva evaluasi model.
* Disimpan sebagai file gambar "roc\_test.png" dan "pr\_test.png".

1. Langkah 6 – Pentingnya Fitur

Kode nya sesuai dari modul Lembar Kerja Pertemuan 6



Output nya:

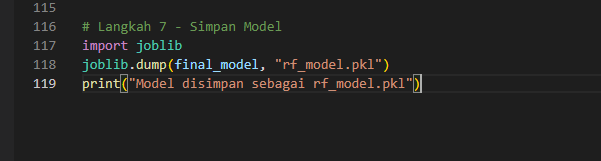


Penjelasan:

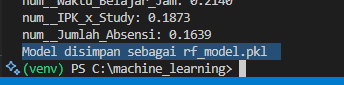
* Menampilkan kontribusi relatif tiap fitur terhadap prediksi model.
* feature\_importances\_ hanya tersedia pada model berbasis pohon seperti Random Forest.
* Disortir dari yang paling penting ke paling kecil.

1. Langkah 7 – Simpan Model

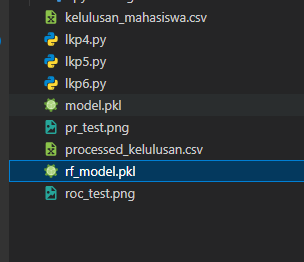
Kode nya:



Output nya:



Menghasilkan file baru rf\_model.pkl:

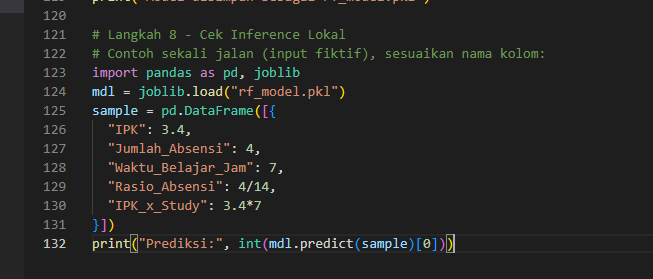


Penjelasan:

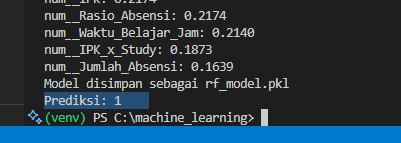
* Model final disimpan dalam file .pkl (pickle) agar bisa digunakan lagi tanpa retrain.
* joblib lebih efisien daripada pickle biasa untuk model besar.

1. Langkah 8 – Cek Inference Lokal

Kode nya:



Output:



Penjelasan:

* Memuat kembali model yang telah disimpan.
* Membuat satu contoh data input (berdasarkan fitur dataset).
* Menampilkan hasil prediksi (misalnya 1 = lulus, 0 = tidak lulus).

Kesimpulan Alur Kerja

| **Langkah** | **Tujuan Utama** | **Hasil** |
| --- | --- | --- |
| 1 | Membaca & membagi data | Dapatkan subset train/val/test |
| 2 | Membangun baseline model | Evaluasi awal performa |
| 3 | Validasi silang | Pastikan model stabil |
| 4 | Tuning parameter | Meningkatkan performa model |
| 5 | Evaluasi akhir | Mengukur performa di test set |
| 6 | Analisis fitur | Mengetahui variabel paling berpengaruh |
| 7 | Simpan model | Bisa digunakan kembali |
| 8 | Prediksi baru | Uji model dengan input nyata |