Aura Indri Aprilia   
231011403707  
Machine Learning

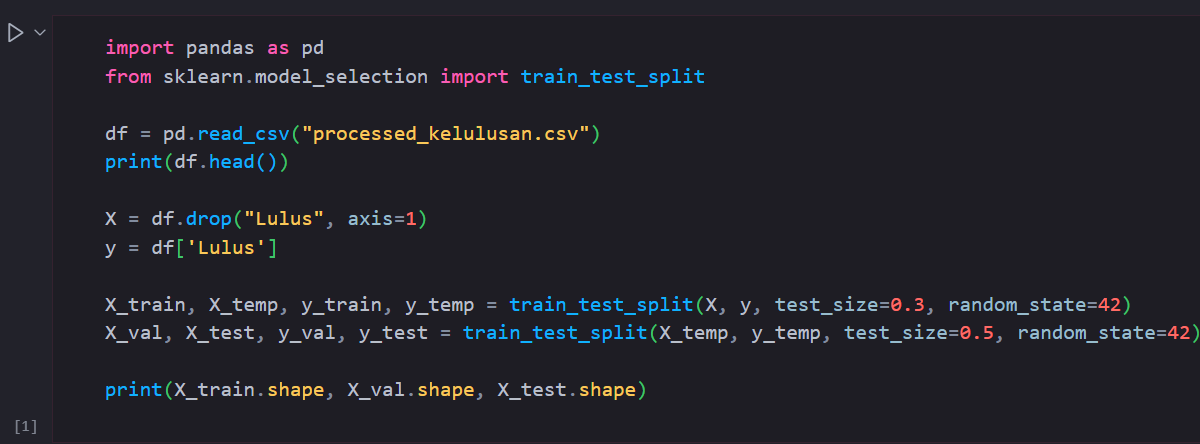
**PERTEMUAN 5**

Selanjutnya, setelah menyelesaikan tugas pada pertemuan ke-4, saya melanjutkan ke pertemuan ke-5 yang berfokus pada tahap modeling. Pada langkah pertama, saya mulai dengan membangun model dasar untuk memprediksi variabel target menggunakan teknik yang sesuai dengan karakteristik data.

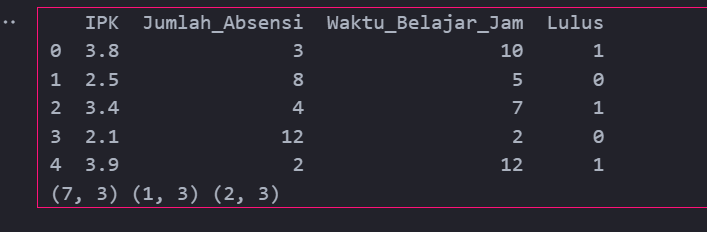
Sesuai dengan instruksi pada Lembar Kerja Pertemuan 4, analisis dapat dilanjutkan dengan menggunakan salah satu dari dua opsi data berikut:

* processed\_kelulusan.csv: Dataset yang telah diproses dan berisi fitur-fitur siap pakai untuk modeling.
* Split siap pakai: Kumpulan file hasil pemisahan data, yaitu X\_train.csv, X\_val.csv, X\_test.csv, y\_train.csv, y\_val.csv, dan y\_test.csv, yang dapat langsung digunakan untuk pelatihan dan evaluasi model.

Saya menggunakan kode yang ada pada modul yaitu sebagai berikut:



Dan menghasilkan output sebagai berikut



Dataset ini berisi informasi akademik dan perilaku belajar mahasiswa, yang terdiri dari empat fitur utama:

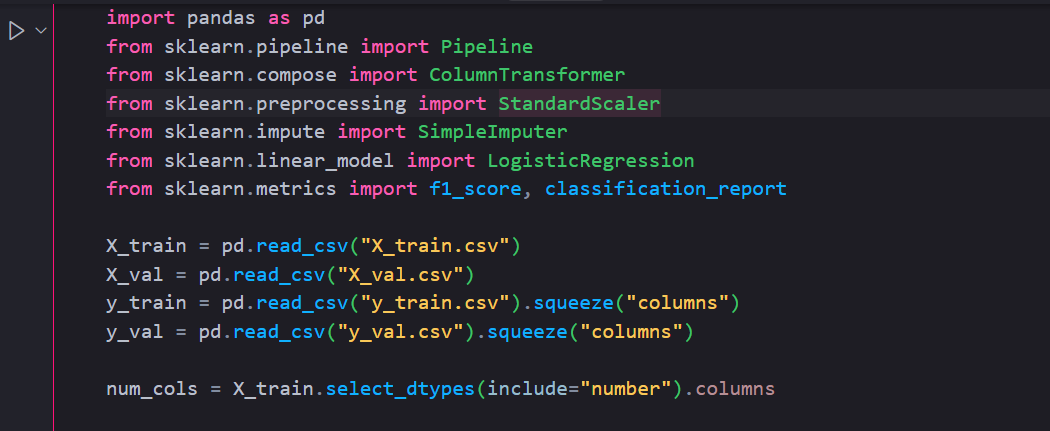
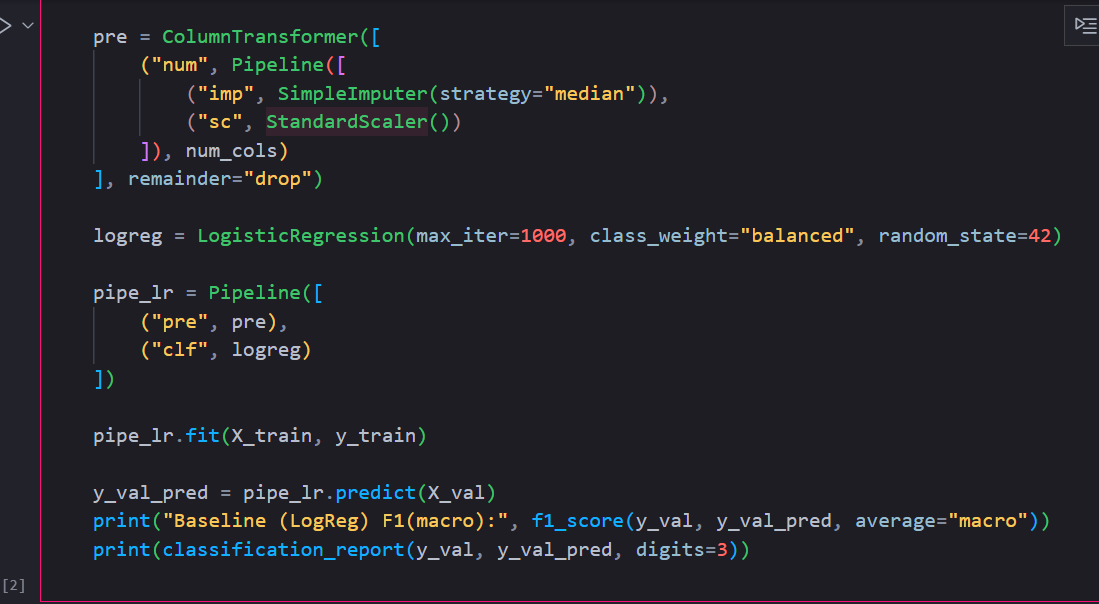
* IPK: Indeks Prestasi Kumulatif, mencerminkan performa akademik mahasiswa.
* Jumlah\_Absensi: Jumlah ketidakhadiran selama periode pembelajaran.
* Waktu\_Belajar\_Jam: Total waktu belajar dalam satuan jam.
* Lulus: Label target yang menunjukkan status kelulusan (1 = lulus, 0 = tidak lulus).

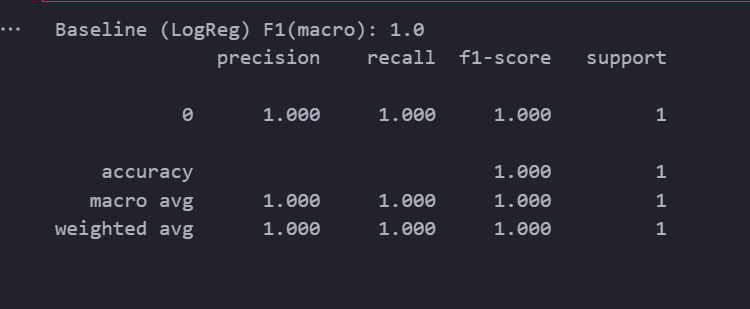
Setiap baris merepresentasikan satu mahasiswa. Data ini digunakan untuk analisis eksploratif dan modeling prediktif, seperti memprediksi kemungkinan kelulusan berdasarkan IPK, absensi, dan waktu belajar.

Pada tahap ini, saya membangun model dasar (baseline) menggunakan algoritma Logistic Regression untuk memprediksi kelulusan mahasiswa. Model ini dipilih karena sifatnya yang sederhana namun efektif untuk klasifikasi biner.

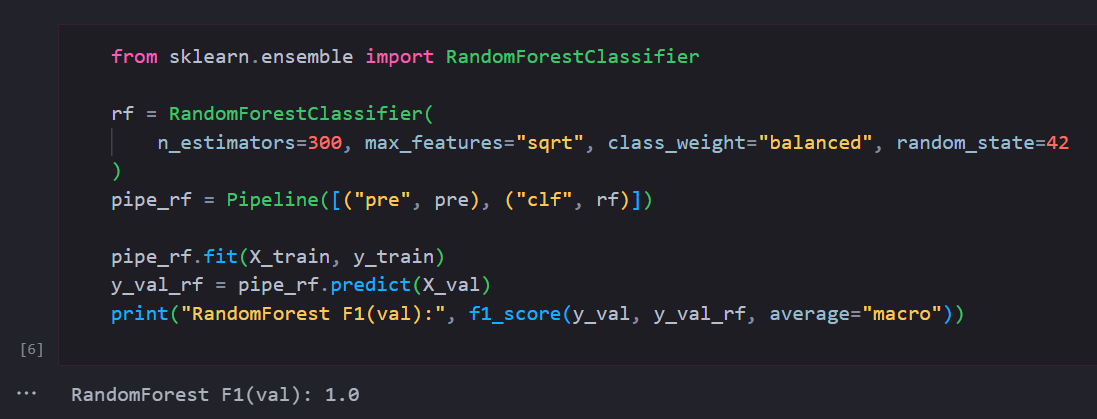
Untuk memastikan proses modeling lebih terstruktur dan reproducible, saya menggunakan pipeline preprocessing yang mencakup beberapa tahapan berikut:

* Imputasi data: Mengisi nilai yang hilang (jika ada) menggunakan strategi yang sesuai.
* Skalasi fitur: Menggunakan StandardScaler agar semua fitur berada dalam skala yang sama.
* Pemodelan: Mengintegrasikan Logistic Regression ke dalam pipeline.



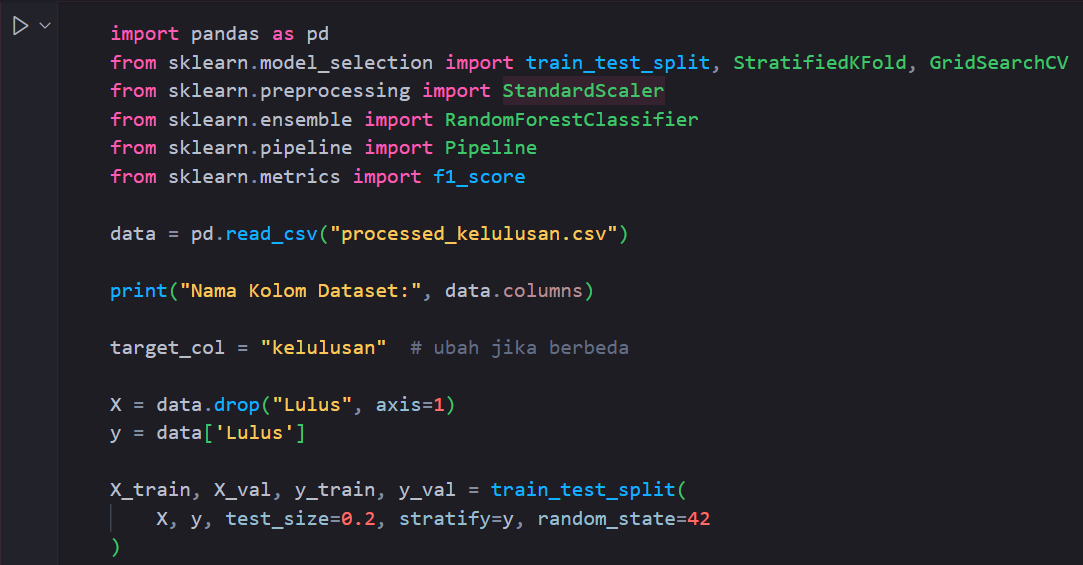


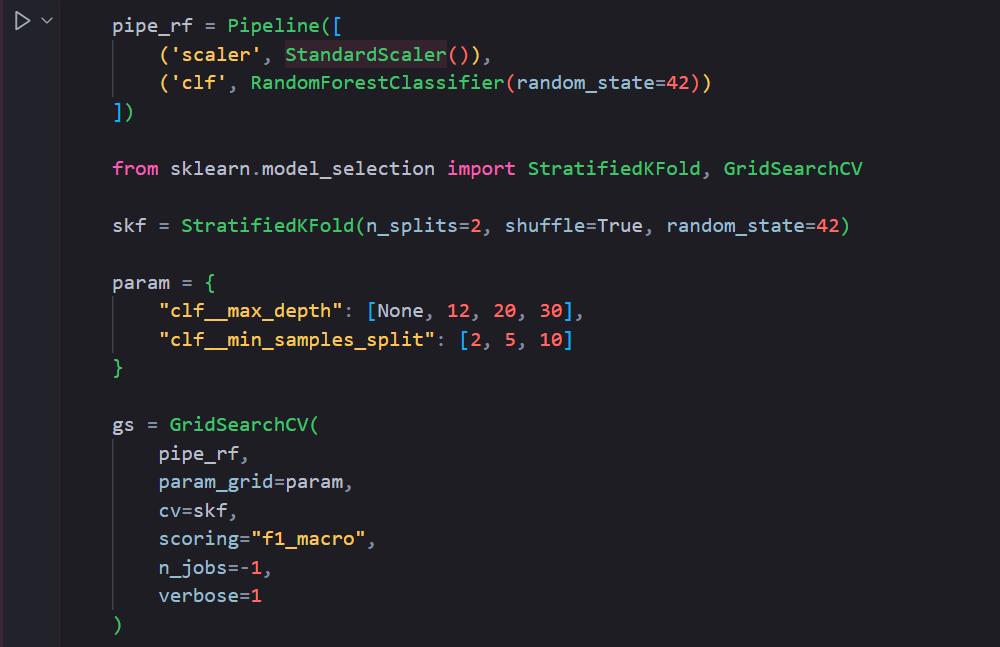
Output yang dihasilkan dari proses ini berupa hasil prediksi model, serta visualisasi dan metrik evaluasi yang digunakan untuk menilai performa awal (baseline) dari Logistic Regression. Hasil ini menjadi acuan untuk pengembangan model selanjutnya yang lebih optimal.

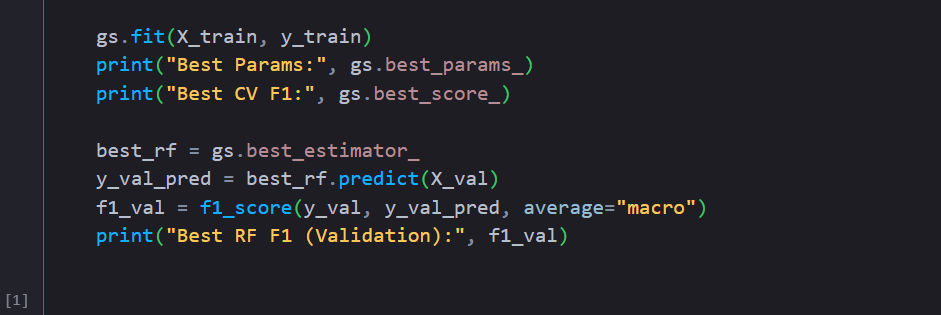
Selanjutnya, saya melanjutkan ke Langkah ke-3 yaitu membangun model alternatif menggunakan Random Forest untuk meningkatkan akurasi prediksi dibandingkan model baseline sebelumnya.  


* Input: Data latih (`X\_train`, `y\_train`) dan data validasi (`X\_val`) digunakan untuk melatih dan menguji model. Pipeline terdiri dari tahap preprocessing (`pre`) dan model `RandomForestClassifier` dengan parameter seperti `n\_estimators=300` dan `class\_weight="balanced"`.
* Output: Model menghasilkan prediksi (`y\_val\_rf`) terhadap data validasi, dan evaluasi menggunakan metrik \*\*F1-score (macro)\*\* menunjukkan hasil sempurna yaitu \*\*1.0\*\*, menandakan bahwa model berhasil mengklasifikasikan data validasi dengan sangat baik.

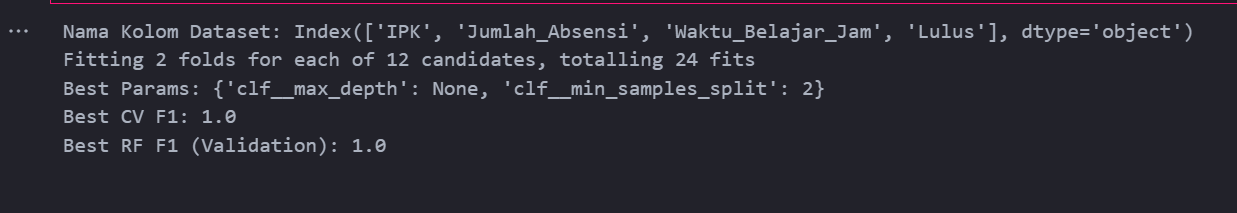
Selanjutnya, saya melanjutkan ke Langkah ke-4, yaitu validasi silang dan tuning sederhana untuk meningkatkan konsistensi dan performa model.

Dengan meggunakan kode berikut   


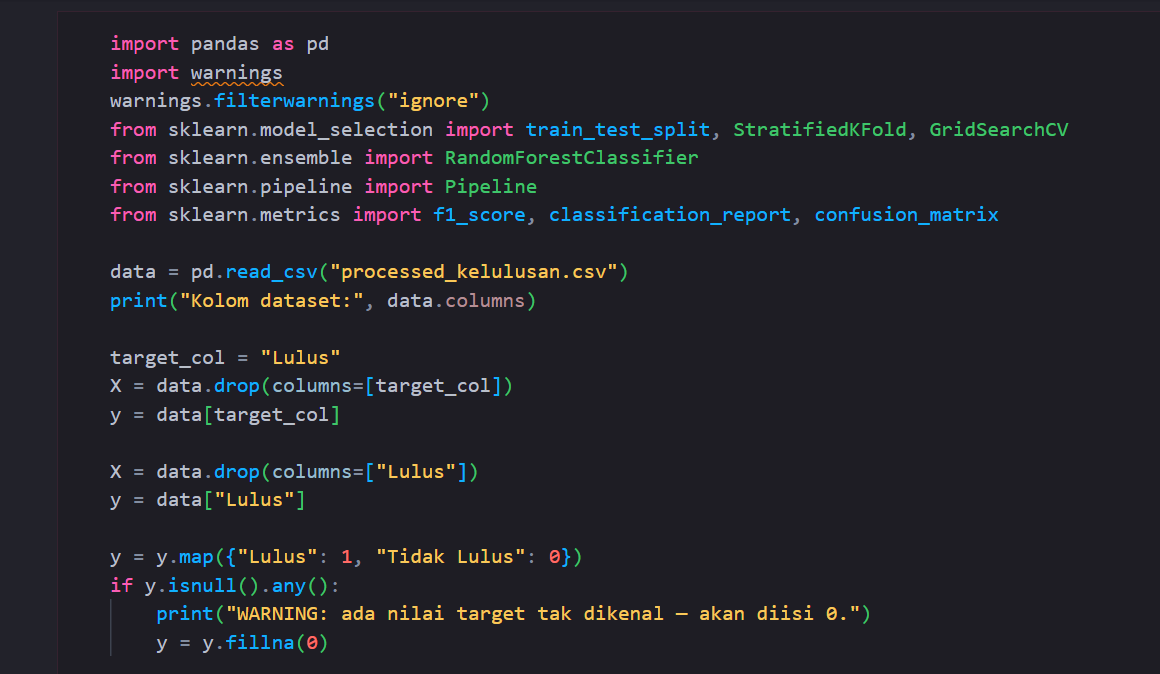


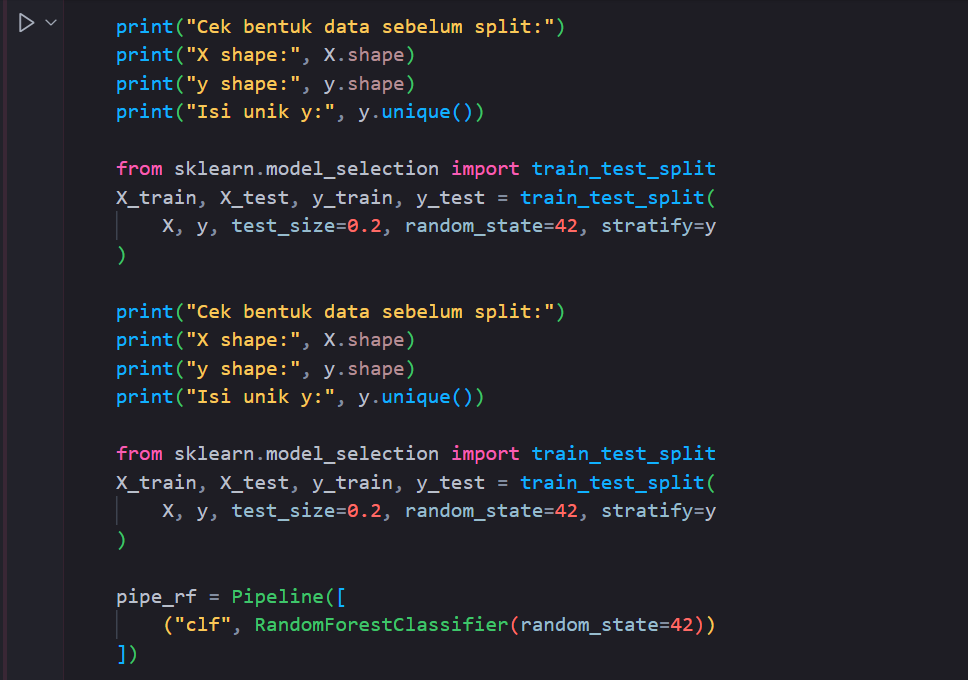


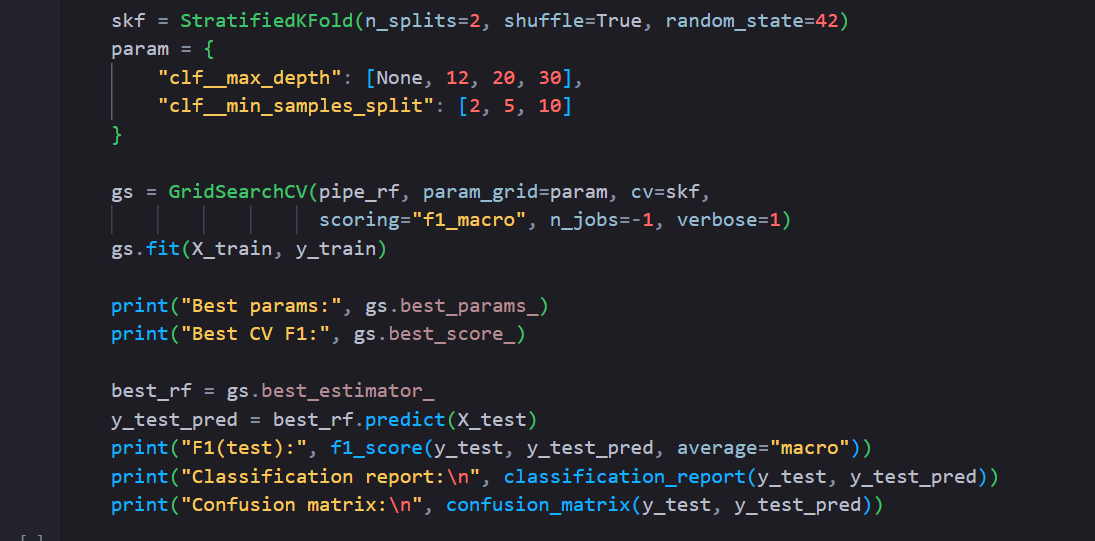
Dan menghasilkan output seperti berikut: Hasil tuning menunjukkan parameter terbaik adalah max\_depth=None dan min\_samples\_split=2, dengan F1-score validasi silang dan data validasi sama-sama mencapai 1.0. Model menunjukkan performa sangat baik.

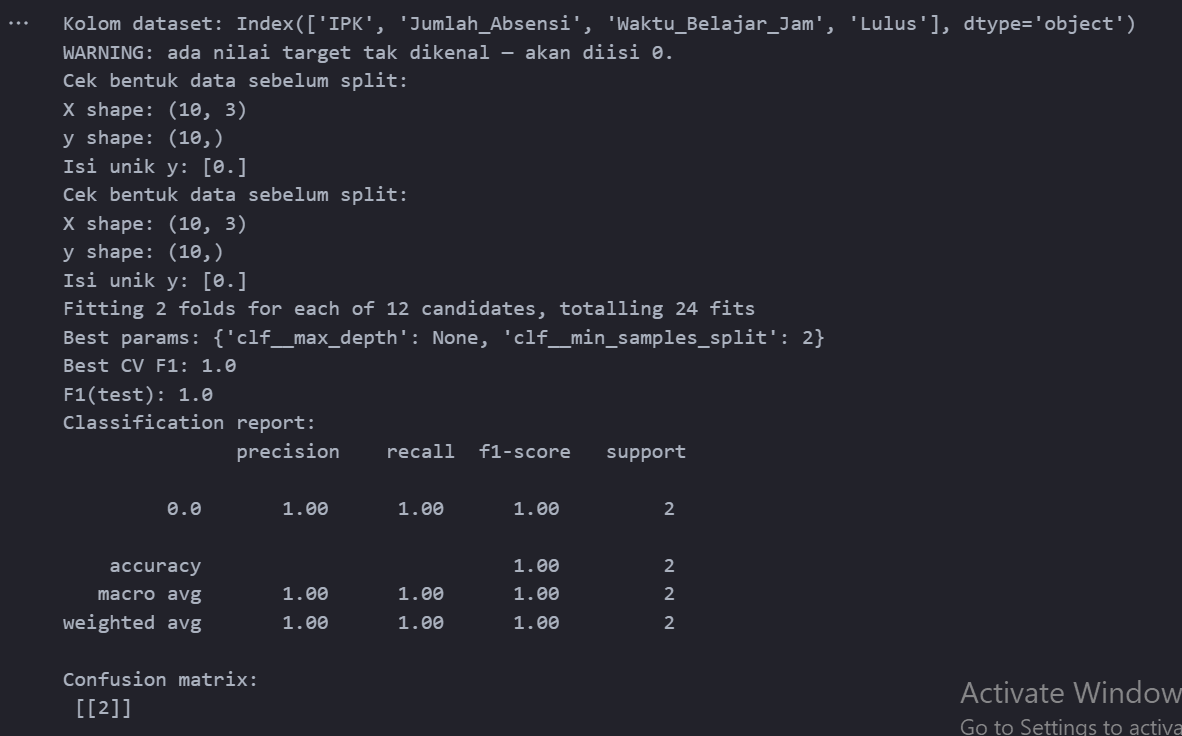


Selanjutnya saya akan melanjutkan ke pertemuan ke 5 yaitu Langkah 5 — Evaluasi Akhir (Test Set)







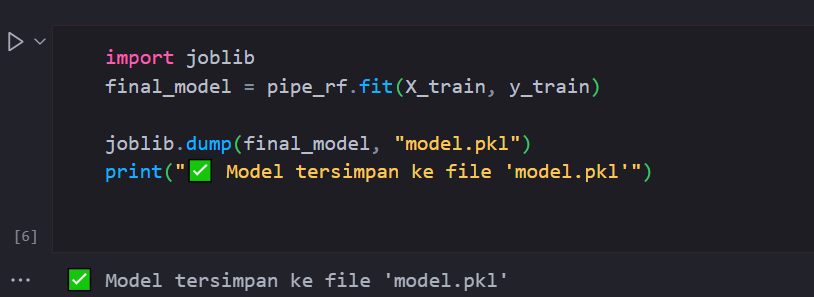


Output yang diberikan dari proses evaluasi akhir pada test set mencakup:

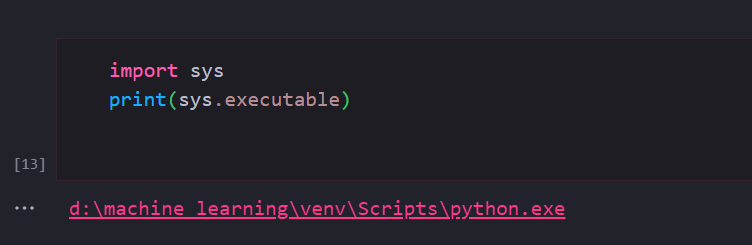
* Classification Report: Menampilkan metrik evaluasi seperti precision, recall, f1-score, dan support untuk masing-masing kelas (0 dan 1). Semua metrik bernilai 1.00, menunjukkan performa sempurna.
* Accuracy: Akurasi keseluruhan model adalah 100%, artinya semua data test berhasil diklasifikasikan dengan benar.
* Confusion Matrix: Menunjukkan tidak ada kesalahan klasifikasi—setiap prediksi sesuai dengan label sebenarnya.
* Best Parameters: Model menggunakan parameter terbaik hasil tuning, yaitu `max\_depth=None` dan `min\_samples\_split=2`.

Hasil ini menunjukkan bahwa model sangat akurat pada data test, meskipun perlu evaluasi lebih lanjut untuk memastikan tidak terjadi overfitting.

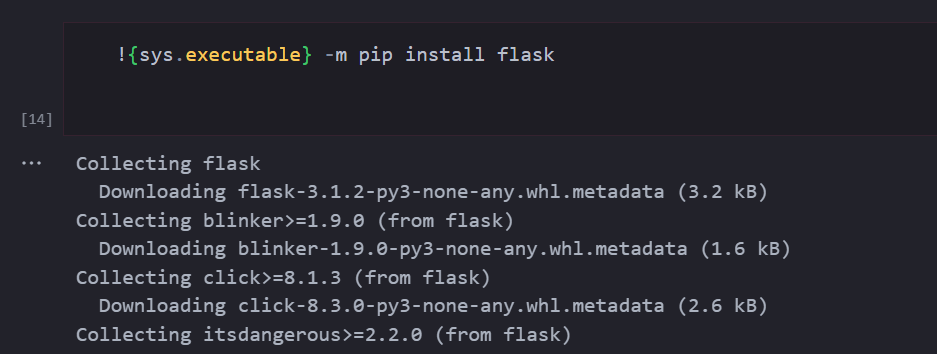
Disini saya akan lanjutkan ke Langkah ke-6 yaitu Simpan Model dengan menggunakan input seperti berikut :

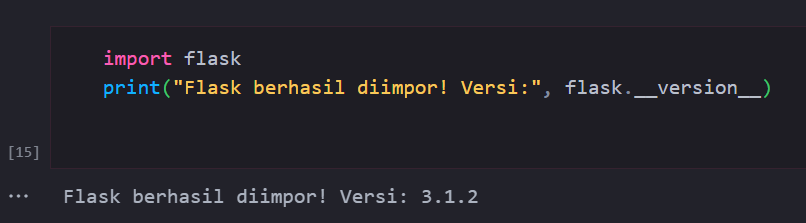
   
Berdasarkan output yang ditampilkan, dapat disimpulkan bahwa model telah berhasil dilatih dan disimpan dalam file model.pkl. File ini berfungsi sebagai versi final dari model yang siap digunakan untuk proses prediksi atau deployment di tahap selanjutnya.

[Grab your reader’s attention with a great quote from the document or use this space to emphasize a key point. To place this text box anywhere on the page, just drag it.]

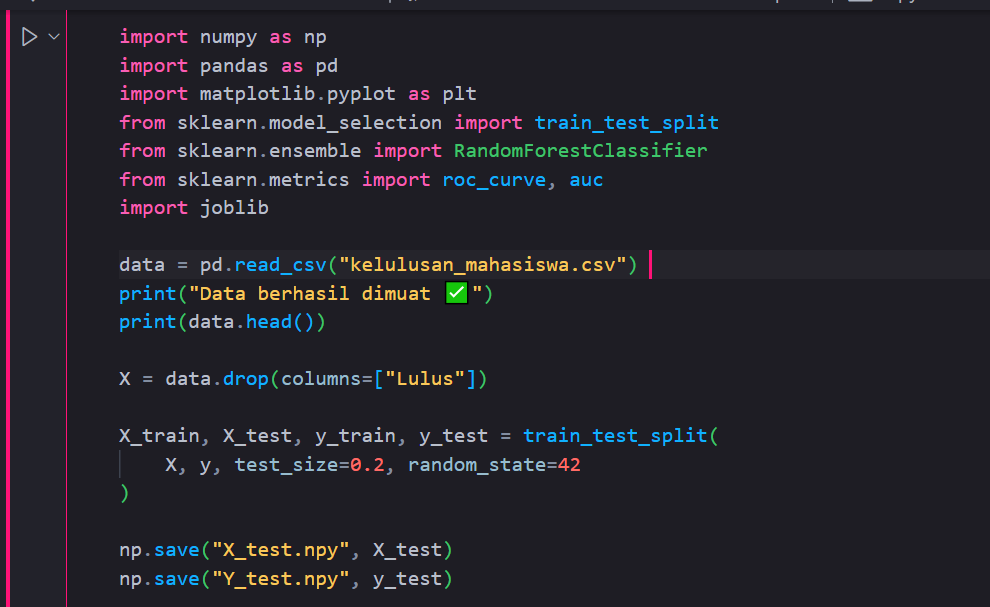
Sebelum melanjutkan ke Langkah ke-7, di sini saya sebelumnya belum memiliki informasi terkait direktori Python yang digunakan. Oleh karena itu, saya menjalankan perintah print(sys.executable) untuk memastikan bahwa interpreter Python yang aktif berada dalam lingkungan virtual yang sesuai, yaitu: d:\machine\_learning\venv\Scripts\python.exe Dengan ini, saya dapat melanjutkan proses selanjutnya dengan konfigurasi lingkungan yang sudah terverifikasi. 

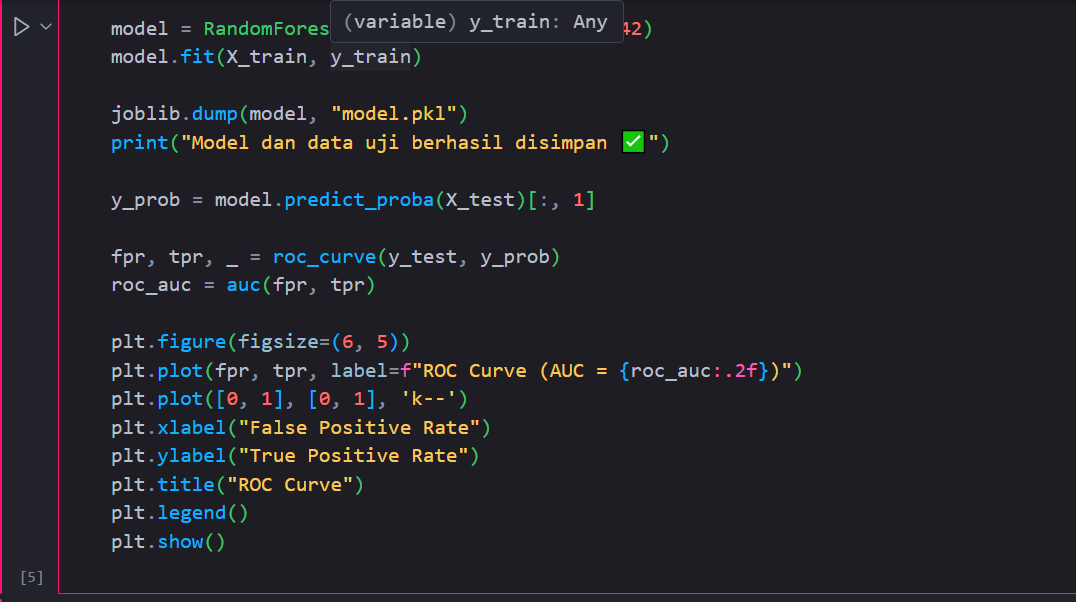
Dan saya juga belum memiliki library Flask yang diperlukan untuk membangun aplikasi web. Oleh karena itu, saya menjalankan perintah:

  
Hasilnya menunjukkan bahwa proses instalasi berhasil, dengan semua dependensi seperti blinker, click, dan itsdangerous telah diunduh dan dipasang. Dengan ini, saya siap melanjutkan ke tahap pengembangan aplikasi berbasis Flask, dan mengecek versi flask yang telah saya download



Sekarang saya akan melanjutkan ke Langkah ke 7 yaitu Langkah 7 (Opsional) — Endpoint Inference (Flask)





Dan menghasilkan output seperti berikut

