

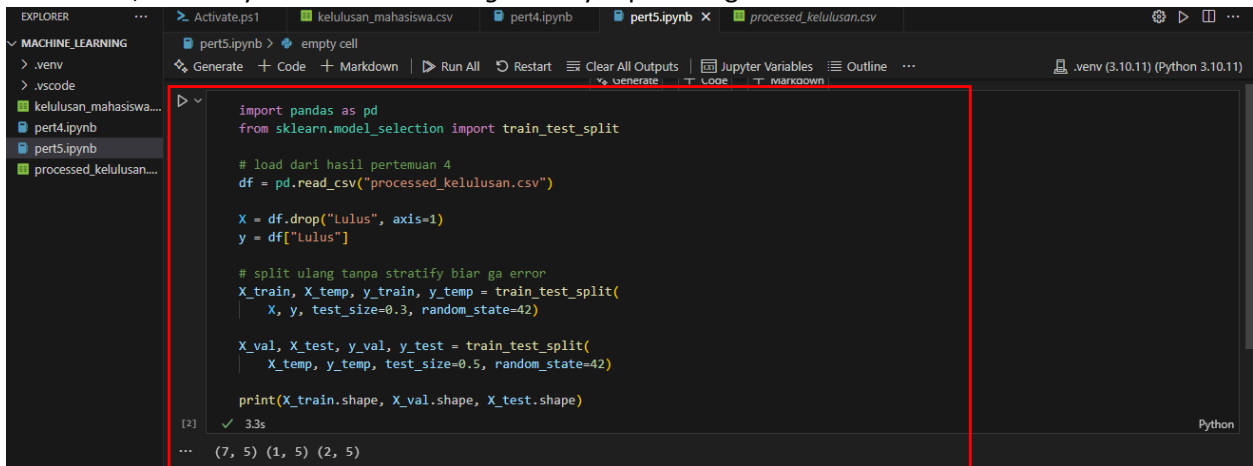
DOKUMENTASI MACHINE LEARNING PERTEMUAN 5

Muhammad Ridho Irfani | 231011402243

05TPLE017

Melanjutkan modul pertemuan 4, sekarang saya akan mengerjakan modul pertemuan 5 dengan materi **Modeling Machine Learning**. Saya langsung membagikan dokumentasi meneruskan pertemuan 4 tapi saya buat file .ipynb baru untuk pertemuan 5 ini sebagai berikut.

1. **Muat data**, disini saya ambil data *existing* lalu saya split ulang.



```
import pandas as pd
from sklearn.model_selection import train_test_split

# load dari hasil pertemuan 4
df = pd.read_csv("processed_kelulusan.csv")

X = df.drop("Lulus", axis=1)
y = df["Lulus"]

# split ulang tanpa stratify biar ga error
X_train, X_temp, y_train, y_temp = train_test_split(
    X, y, test_size=0.3, random_state=42)


X_val, X_test, y_val, y_test = train_test_split(
    X_temp, y_temp, test_size=0.5, random_state=42)

print(X_train.shape, X_val.shape, X_test.shape)
```

[2] ✓ 33s

... (7, 5) (1, 5) (2, 5)

2. **Baseline model & pipeline(logistic regression)**



```
(num, Pipeline([
    ("imp", SimpleImputer(strategy="median")),
    ("sc", StandardScaler())
]), num_cols)
], remainder="drop")

logreg = LogisticRegression(max_iter=1000, class_weight="balanced", random_state=42)
pipe_lr = Pipeline([("pre", pre), ("clf", logreg)])

pipe_lr.fit(X_train, y_train)
y_val_pred = pipe_lr.predict(X_val)

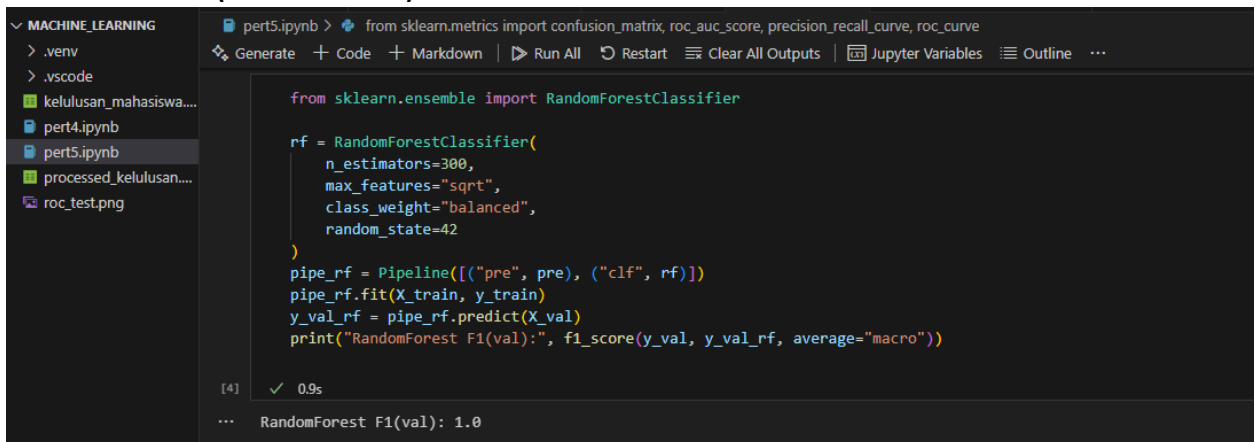
print("Baseline (LogReg) F1(val):", f1_score(y_val, y_val_pred, average="macro"))
print(classification_report(y_val, y_val_pred, digits=3))
```

[3] ✓ 0.5s

... Baseline (LogReg) F1(val): 1.0

	precision	recall	f1-score	support
0	1.000	1.000	1.000	1
accuracy			1.000	1
macro avg	1.000	1.000	1.000	1
weighted avg	1.000	1.000	1.000	1

3. Model Alternatif (Random forest).



```
from sklearn.metrics import confusion_matrix, roc_auc_score, precision_recall_curve, roc_curve

from sklearn.ensemble import RandomForestClassifier

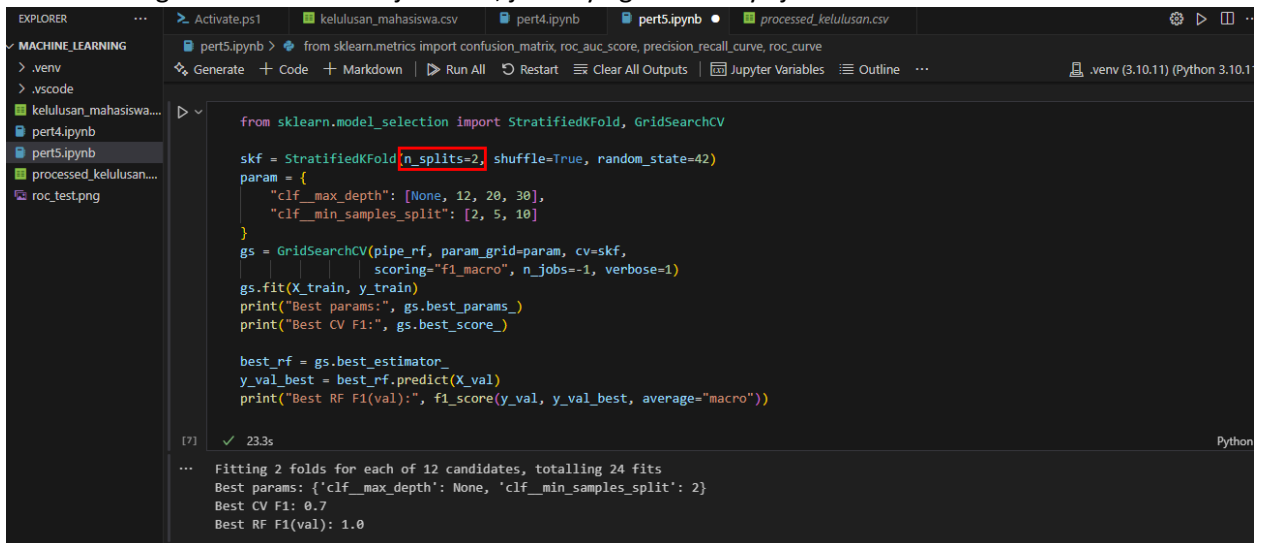
rf = RandomForestClassifier(
    n_estimators=300,
    max_features="sqrt",
    class_weight="balanced",
    random_state=42
)

pipe_rf = Pipeline([("pre", pre), ("clf", rf)])
pipe_rf.fit(X_train, y_train)
y_val_rf = pipe_rf.predict(X_val)
print("RandomForest F1(val):", f1_score(y_val, y_val_rf, average="macro"))
```

[4] ✓ 0.9s

... RandomForest F1(val): 1.0

4. **Validasi Silang & Tuning Ringkas** di langkah ini saya menemukan error ketika mengikuti codingan modul. Dan setelah saya cari tau permasalahannya dikarenakan dataset yang saya miliki terlalu kecil, validasi silang 5-fold tidak bisa dijalankan, jadi saya ganti foldnya jadi 2



```
from sklearn.model_selection import StratifiedKFold, GridSearchCV

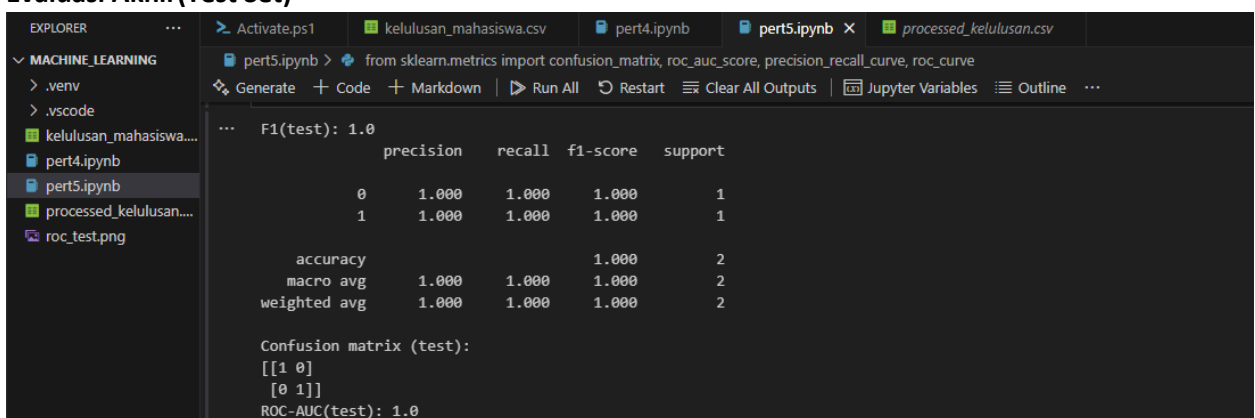
skf = StratifiedKFold(n_splits=2, shuffle=True, random_state=42)
param = {
    "clf__max_depth": [None, 12, 20, 30],
    "clf__min_samples_split": [2, 5, 10]
}
gs = GridSearchCV(pipe_rf, param_grid=param, cv=skf,
                  scoring="f1_macro", n_jobs=-1, verbose=1)
gs.fit(X_train, y_train)
print("Best params:", gs.best_params_)
print("Best CV F1:", gs.best_score_)

best_rf = gs.best_estimator_
y_val_best = best_rf.predict(X_val)
print("Best RF F1(val):", f1_score(y_val, y_val_best, average="macro"))
```

[7] ✓ 233s

... Fitting 2 folds for each of 12 candidates, totalling 24 fits
Best params: {'clf__max_depth': None, 'clf__min_samples_split': 2}
Best CV F1: 0.7
Best RF F1(val): 1.0

5. Evaluasi Akhir(Test Set)

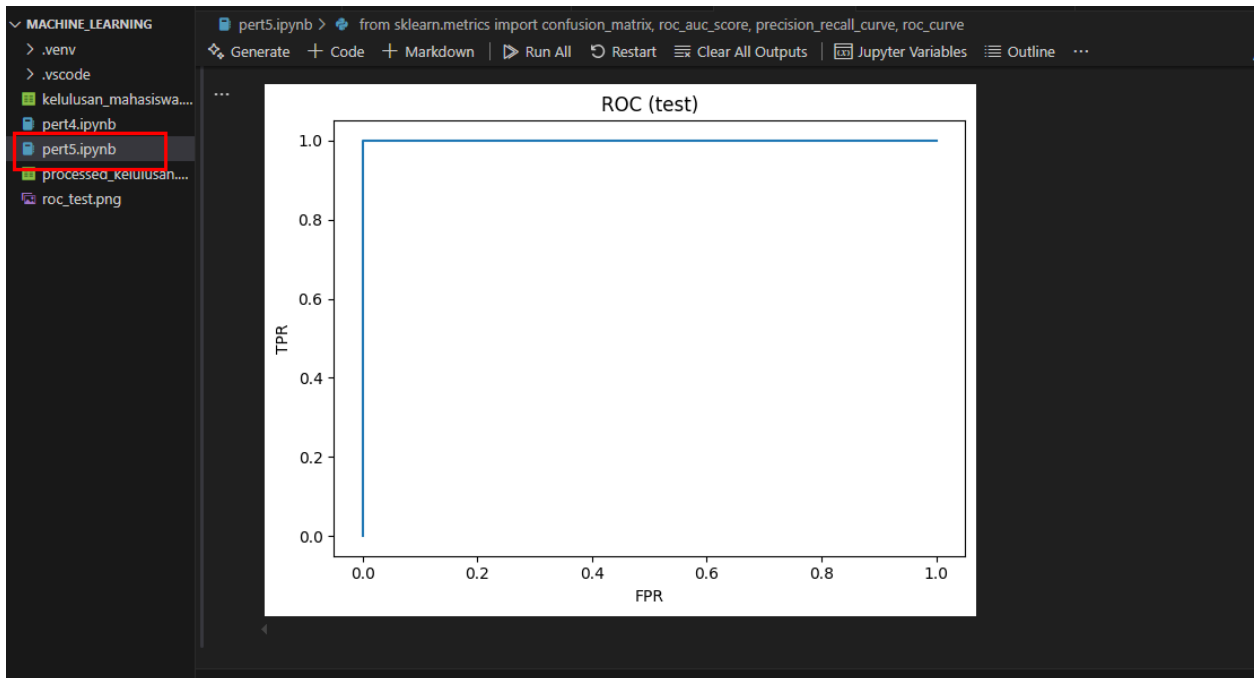


```
from sklearn.metrics import confusion_matrix, roc_auc_score, precision_recall_curve, roc_curve
```

... F1(test): 1.0

	precision	recall	f1-score	support
0	1.000	1.000	1.000	1
1	1.000	1.000	1.000	1
accuracy			1.000	2
macro avg	1.000	1.000	1.000	2
weighted avg	1.000	1.000	1.000	2

Confusion matrix (test):
[[1 0]
[0 1]]
ROC-AUC(test): 1.0



6. Simpan model(optional)

```
import joblib
joblib.dump(final_model, "model.pkl")
print("Model tersimpan ke model.pkl")
```

[9] ✓ 0.2s

Model tersimpan ke model.pkl

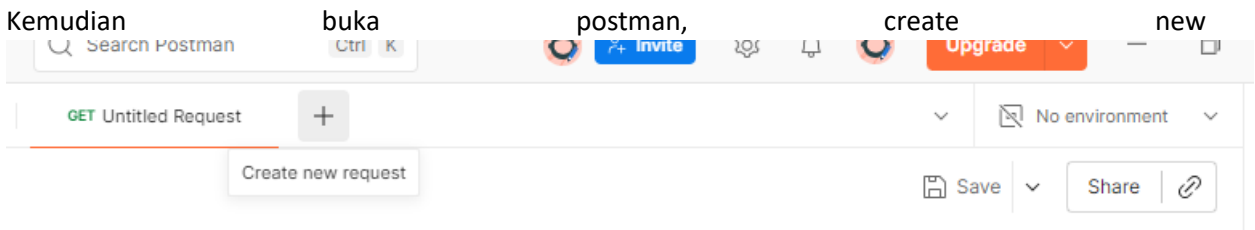
7. Endpoint Inference (flask)

Disini saya ikut mencoba, yaitu saya buat folder baru untuk langkah ini

Kemudian saya masukkan codingan saya. Lalu saya jalankan **python .\step7pert5.py** di dalam terminal, dan muncul pop up seperti ini

```
(.venv) PS D:\machine_learning> python .\step7pert5.py
* Serving Flask app 'step7pert5'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Lalu saya mencoba kembali untuk mencoba mengakses halaman tersebut. Disini saya menggunakan postman, yang bisa di download melalui link resmi postman nya <https://postman.com/downloads/> Kemudian



Lalu isikan seperti gambar di bawah ini. Kemudian bisa kita lihat dibagian paling bawah, preview muncul seperti itu.

The screenshot shows a REST client interface with the following details:

- Request Method:** POST (highlighted with a red box)
- Request URL:** http://127.0.0.1:5000/predict (highlighted with a red box)
- Body Tab:** Selected (highlighted with a red box)
- Body Type:** raw (highlighted with a red box)
- Body Content:** A JSON object (highlighted with a red box):

```
1 {  
2   "IPK": 3.4,  
3   "Jumlah_Absensi": 4,  
4   "Waktu_Belajar_Jam": 7,  
5   "Rasio_Absensi": 0.28,  
6   "IPK_x_Study": 23.8  
7 }  
8
```
- Response Status:** 200 OK
- Response Headers:** 96 ms, 196 B
- Response Body:** A JSON object:

```
{  
  "prediction": 1,  
  "proba": 0.97  
}
```