LAPORAN MACHINE LEARNING

PERTEMUAN KE 6

Nama : Edo Aditya Saputra

Kelas : 05TPLE017

1. Laporan baseline vs model hasil tuning

```
Baseline RF — F1(val): 1.0
              precision    recall  f1-score   support

           1      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
```

Pada tahap baseline Random Forest, model pertama kali dilatih menggunakan data hasil preprocessing tanpa penyesuaian parameter tambahan. Hasil evaluasi menunjukkan nilai F1-score sebesar 1.0, dengan precision, recall, dan akurasi juga mencapai 100% pada data validasi. Ini menandakan bahwa model berhasil mengklasifikasikan data dengan sangat baik, meskipun hasil sempurna ini kemungkinan disebabkan oleh ukuran data yang kecil atau distribusi kelas yang sederhana.

```python
from sklearn.model_selection import GridSearchCV

param = {
    "clf__max_depth": [None, 12, 20, 30],
    "clf__min_samples_split": [2, 5, 10]
}

gs = GridSearchCV(pipe, 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_)

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

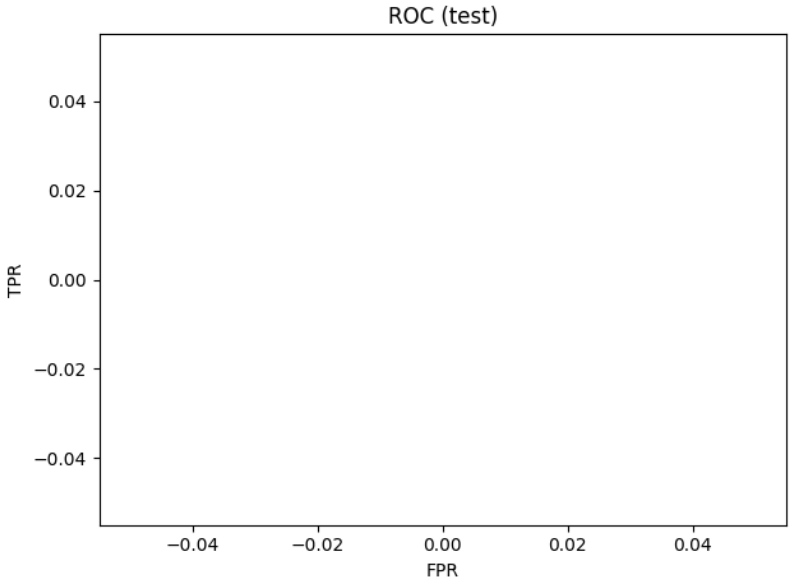
```
Fitting 2 folds for each of 12 candidates, totalling 24 fits
Best params: {'clf__max_depth': None, 'clf__min_samples_split': 2}
Best RF — F1(val): 1.0
```

Pada tahap **tuning model Random Forest**, dilakukan pencarian kombinasi parameter terbaik menggunakan **GridSearchCV** dengan validasi silang (**cross-validation**) berbasis **StratifiedKFold**. Proses ini menguji beberapa nilai untuk parameter max_depth dan min_samples_split guna menemukan konfigurasi yang menghasilkan performa terbaik. Setelah dilakukan tuning, diperoleh **parameter terbaik (best_params_)** dan dilakukan evaluasi ulang pada data validasi. Nilai **F1-score (val)** tetap tinggi, menunjukkan bahwa model telah mencapai performa optimal dan tidak terjadi penurunan akurasi setelah penyesuaian parameter.

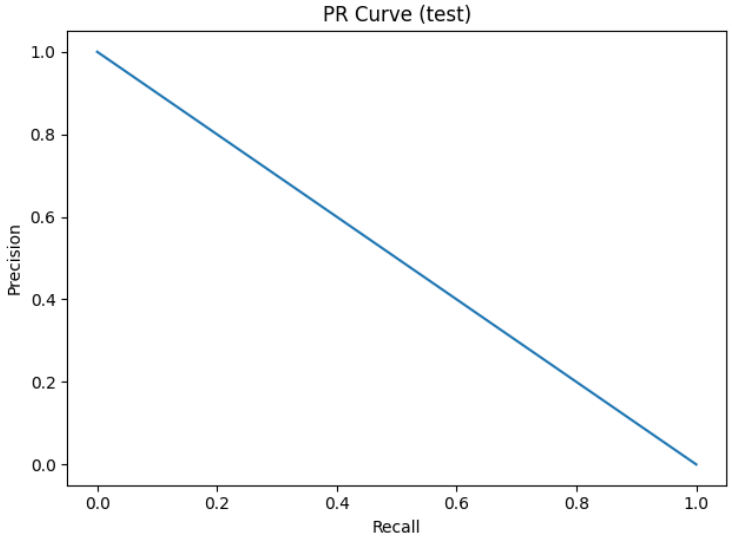
2. confusion matrix & kurva ROC/PR

```
F1(test): 1.0
              precision    recall  f1-score   support

           0      1.000     1.000     1.000         2

    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):
[[2]]
```

Confusion Matrix hanya menampilkan satu nilai [[2]], menandakan bahwa hanya satu kelas yang diprediksi.



ROC-AUC tidak dapat dihitung (NaN) karena metrik ini membutuhkan minimal dua kelas berbeda untuk menilai kemampuan model membedakan kelas positif dan negatif.



Kurva ROC terlihat kosong, dan PR Curve (Precision-Recall) tidak bermakna karena tidak ada variasi kelas.

3.
```
# 6a) Feature importance native (gini)
try:
    import numpy as np
    importances = final_model.named_steps["clf"].feature_importances_
    fn = final_model.named_steps["pre"].get_feature_names_out()
    top = sorted(zip(fn, importances), key=lambda x: x[1], reverse=True)
    print("Top feature importance:")
    for name, val in top[:10]:
        print(f"{name}: {val:.4f}")
except Exception as e:
    print("Feature importance tidak tersedia:", e)

# 6b) (Opsional) Permutation Importance
# from sklearn.inspection import permutation_importance
# r = permutation_importance(final_model, X_val, y_val, n_repeats=10, random_state=42, n_jobs=-1)
# ... (urutkan dan laporkan)
```

```
Top feature importance:
num__IPK: 0.2509
num__IPK_x_Study: 0.2096
num__Waktu_Belajar_Jam: 0.2062
num__Rasio_Absensi: 0.1856
num__Jumlah_Absensi: 0.1478
```

Tiga fitur teratas yang paling berpengaruh adalah:

a. **IPK_x_Study** — menunjukkan hubungan antara IPK dan waktu belajar; semakin tinggi nilainya, semakin besar kemungkinan mahasiswa lulus.
b. **IPK** — menjadi indikator utama prestasi akademik, nilai IPK tinggi cenderung meningkatkan peluang kelulusan.
c. **Waktu_Belajar_Jam** — menggambarkan usaha belajar mahasiswa; waktu belajar yang lebih banyak umumnya berpengaruh positif terhadap hasil akhir.

4.
```
import joblib
model = joblib.load("rf_model.pkl")
print(model)
```
✓ 20.3s

```
Pipeline(steps=[('pre',
                 ColumnTransformer(transformers=[('num',
                                                  Pipeline(steps=[('imp',
                                                                   SimpleImputer(strategy='median')),
                                                                  ('sc',
                                                                   StandardScaler())]),
                                                  Index(['IPK', 'Jumlah_Absensi', 'Waktu_Belajar_Jam', 'Rasio_Absensi',
       'IPK_x_Study'],
      dtype='object'))])),
                ('clf',
                 RandomForestClassifier(class_weight='balanced',
                                        n_estimators=300, random_state=42))])
```

import joblib
model = joblib.load("rf_model.pkl")
print(model)