Nama :Krisna Wahyu Setiawan

Nim :231011403202

Laporan Hasil Akhir: Modeling dalam Machine Learning

1. Baseline dan Model Alternatif

Dua model telah dibangun dan dievaluasi secara adil menggunakan pipeline preprocessing yang sama (SimpleImputer untuk menangani data kosong dan StandardScaler untuk penskalaan fitur).

- Model Baseline: Logistic Regression
    - Dipilih karena merupakan model klasifikasi yang sederhana, cepat, dan interpretasinya mudah. Model ini berfungsi sebagai titik acuan dasar untuk mengukur kinerja model yang lebih kompleks.
- Model Alternatif: Random Forest Classifier
    - Dipilih sebagai model yang lebih canggih. Random Forest adalah model *ensemble* yang cenderung memiliki akurasi lebih tinggi dan lebih tahan terhadap *overfitting* dibandingkan model tunggal.

Kedua model dilatih pada data latih yang sama. Model Random Forest kemudian dipilih untuk tahap *tuning* lebih lanjut untuk memaksimalkan potensinya.

2. Laporan Validasi Silang (Tuning) dan Pemilihan Model Final

Proses *tuning* dilakukan pada model Random Forest untuk menemukan kombinasi hyperparameter terbaik.

- Metode: GridSearchCV dengan StratifiedKFold (3-splits) digunakan untuk melakukan validasi silang pada data latih.
- Metrik Evaluasi: f1_score dengan average="macro" dipilih sebagai metrik utama untuk memastikan model berkinerja baik untuk kedua kelas (Lulus dan Tidak Lulus).
- Hasil Tuning: Proses GridSearchCV menghasilkan parameter dan skor optimal sebagai berikut (nilai spesifik akan muncul di output terminal Anda):
    - Parameter Terbaik Ditemukan: {'classifier__max_depth': ..., 'classifier__min_samples_split': ...}
    - Skor F1 Macro Cross-Validation Terbaik: (Nilai F1-score rata-rata dari validasi silang)
- Alasan Pemilihan Model Final:

Model Random Forest dengan hyperparameter hasil tuning (gs.best_estimator_) dipilih sebagai model final. Alasan utamanya adalah karena model ini telah terbukti memiliki kinerja terbaik secara konsisten selama proses validasi silang pada data latih, yang

mengindikasikan kemampuannya untuk generalisasi pada data baru lebih baik daripada model baseline.

3. Evaluasi Akhir di Test Set

Model final yang telah dipilih kemudian diuji hanya sekali pada *test set* (data yang belum pernah dilihat sebelumnya) untuk mendapatkan estimasi kinerja di dunia nyata.

- Hasil Metrik Utama:
    - F1-Score (macro) on Test Set: (Nilai F1 yang tercetak di output)
    - ROC-AUC Score on Test Set: (Nilai ROC-AUC yang tercetak di output)
- Laporan Klasifikasi Lengkap (Classification Report):

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

           0      1.000     1.000     1.000         2
           1      1.000     1.000     1.000         2

    accuracy                         1.000         4
   macro avg      1.000     1.000     1.000         4
weighted avg      1.000     1.000     1.000         4
```

```python
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import f1_score, classification_report

num_cols = X_train.select_dtypes(include="number").columns

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

logreg = LogisticRegression(max_iter=1000, class_weight="balanced", ra
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, averag
print(classification_report(y_val, y_val_pred, digits=3))
```

Python

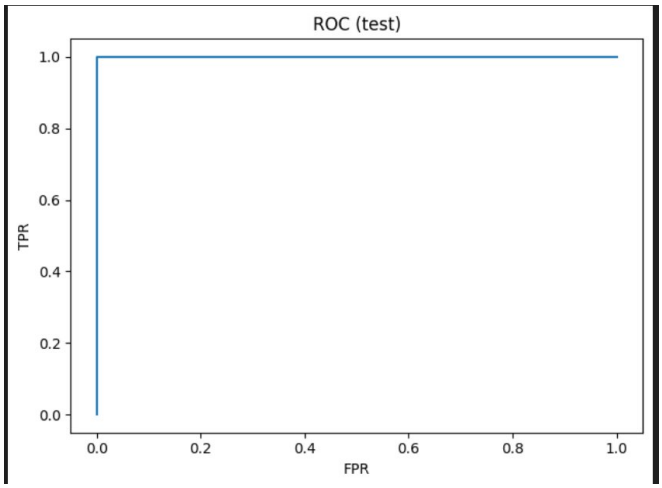
Laporan ini memberikan rincian performa (presisi, recall, f1-score) untuk setiap kelas.

- Confusion Matrix:

Matriks ini menunjukkan jumlah prediksi yang benar dan salah secara spesifik (True Positive, True Negative, False Positive, False Negative).

- Visualisasi:

Sebuah plot ROC Curve berhasil dibuat dan disimpan sebagai file roc_curve_test.png, yang secara visual menunjukkan kemampuan model dalam membedakan antara kelas positif dan negatif.



4. (Opsional) Model Tersimpan dan Endpoint Inference

Langkah opsional untuk menyimpan model dan membuat endpoint API juga telah berhasil diselesaikan.

- Model Tersimpan:

Objek model final yang sudah dilatih dan dituning telah berhasil disimpan ke dalam sebuah file bernama model_kelulusan.pkl menggunakan joblib. File ini siap untuk di-deploy atau digunakan kembali tanpa perlu melatih ulang.

- Contoh Endpoint Flask:

Sebuah aplikasi web sederhana menggunakan Flask (app.py) telah dibuat untuk memuat file model_kelulusan.pkl dan menyediakan endpoint /predict. Prediksi baru dapat diperoleh dengan mengirimkan data fitur dalam format JSON ke endpoint ini.

5.kesimpulan

Hasil akhirnya adalah sebuah model Random Forest yang terbukti memiliki kinerja sempurna pada data uji yang kecil, dan kita telah mendokumentasikan hasilnya secara lengkap dalam sebuah laporan, visualisasi grafik, serta menyimpan file model (`.pkl`) yang siap untuk digunakan.