

Random Forest

Cara Kerja:

1. Bentuk subset data dengan teknik *bootstrap sampling*.
2. Bentuk *Decision Tree* dari subset data yang telah dibuat.
3. Tentukan fitur yang ingin digunakan pada setiap pohon yang digunakan.
4. Lakukan pelatihan pada pohon yang telah dibuat sebelumnya.
5. Lakukan prediksi menggunakan pohon tersebut.
6. Gabungkan hasil dari banyak pohon berbeda dan pilih pohon yang terbaik.

Perbandingan:

- *Scratch*

```
[47]: from randomForest import RandomForestClassifierScratch
```

```
rf_scratch = RandomForestClassifierScratch()
rf_scratch.fit(X_train, y_train)
y_pred_rf_scratch = rf_scratch.predict(X_test)

validate_model(rf_scratch, method_name="Random Forest from Scratch")
```

Hold-Out Validation (Random Forest from Scratch):

F1 Score: 0.9382716049382716

	precision	recall	f1-score	support
0	0.95	0.99	0.97	72
1	0.97	0.90	0.94	42
accuracy			0.96	114
macro avg	0.96	0.95	0.95	114
weighted avg	0.96	0.96	0.96	114

K-Fold Cross-Validation (Random Forest from Scratch):

F1 Scores for each fold: [0.9696969696969697, 0.96875, 0.9166666666666666, 0.927536231884058, 0.8709677419354839]

Mean F1 Score: 0.9307235220366357

Standard Deviation of F1 Score: 0.036724514716295946

- *Library*

```
[48]: from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(n_estimators=100, max_depth=None)
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)

validate_model(rf, method_name="Random Forest from Library")
```

Hold-Out Validation (Random Forest from Library):
F1 Score: 0.9629629629629629

	precision	recall	f1-score	support
0	0.96	1.00	0.98	72
1	1.00	0.93	0.96	42
accuracy			0.97	114
macro avg	0.98	0.96	0.97	114
weighted avg	0.97	0.97	0.97	114

K-Fold Cross-Validation (Random Forest from Library):
F1 Scores for each fold: [1.0, 0.9846153846153847, 0.9014084507042254, 0.9714285714285714, 0.8709677419354839]
Mean F1 Score: 0.945684029736733
Standard Deviation of F1 Score: 0.05034184543044354

Dari implementasi secara *scratch* dan *library*, terlihat bahwa penggunaan *library* memiliki nilai F1 score yang lebih tinggi. Ini semakin dibuktikan dengan presisi memprediksi diagnosis kanker(1) dengan skor 1.

Improvement:

Improvement yang dapat dilakukan pada algoritma *Random Forest* secara *scratch* dapat dilakukan dengan memperbaiki *max_features*, jumlah pohon yang digunakan, dan kedalaman maksimal yang diinginkan.