

1. Jelaskan cara kerja dari algoritma tersebut!

Jawab:

Random Forest adalah algoritma *ensemble* yang menggunakan banyak *decision tree* untuk meningkatkan akurasi dan mengurangi *overfitting*. Algoritma ini melatih beberapa *decision tree* pada subset acak dari data dan fitur, kemudian menggabungkan prediksinya. Cara kerjanya adalah sebagai berikut:

- Data dibagi menjadi beberapa subset acak dengan *replacement*.
- Pembentukan Decision Tree, setiap *tree* dilatih menggunakan subset data dan subset fitur acak.
- Setiap *tree* membuat prediksi. Untuk klasifikasi, hasil ditentukan dengan *majority voting*. Untuk regresi, hasil diambil rata-ratanya.
- Pada setiap node, fitur dipilih secara acak untuk mengurangi *overfitting*.

4. Bandingkan hasil evaluasi model pada nomor 2 dan 3, bagaimana hasil perbandingannya? Jika ada perbedaan, jelaskan alasannya!

Jawab:

Menggunakan Implementasi From Scratch

```
import random_forest
importlib.reload(random_forest)

from random_forest import RandomForest

rf = RandomForest(n_trees=100, max_depth=5)

# Using holdout validation
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
ho_accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {ho_accuracy}")

# Using k-fold cross-validation
cv_accuracy = cross_val_score(rf, X, y, cv=kfold, scoring="accuracy").mean()
print(f"Cross-validation Accuracy: {cv_accuracy}")
```

✓ 1m 12.3s

Accuracy: 0.7049180327868853

Cross-validation Accuracy: 0.7519125683060109

Menggunakan Implementasi dari Library

```
rf_lib = RandomForestClassifier(n_estimators=100, max_depth=5, random_state=42)

# Using holdout validation
rf_lib.fit(X_train, y_train)
y_pred_lib = rf_lib.predict(X_test)
ho_accuracy_lib = accuracy_score(y_test, y_pred_lib)
print(f"Accuracy: {ho_accuracy_lib}")

# Using k-fold cross-validation
cv_accuracy_lib = cross_val_score(rf_lib, X, y, cv=kfold, scoring="accuracy").mean()
print(f"Cross-validation Accuracy: {cv_accuracy_lib}")
```

✓ 0.7s

Accuracy: 0.7540983606557377

Cross-validation Accuracy: 0.7914754098360656

Kedua model memberikan hasil akurasi yang mirip pada holdout validation dan sangat mendekati pada cross-validation. Perbedaan kecil yang terlihat bisa disebabkan oleh perbedaan optimasi internal pada library, tetapi secara umum, keduanya menghasilkan performa yang serupa.

5. Jelaskan improvement apa saja yang bisa Anda lakukan untuk mencapai hasil yang lebih baik dibandingkan dengan hasil yang Anda punya saat ini! Improvement yang dimaksud tidak terbatas pada bagaimana algoritma diimplementasikan, namun juga mencakup tahap sebelum modeling and validation.

Jawab:

- Tambahkan atau transformasi fitur untuk menonjolkan pola yang lebih relevan dalam data.
- Melakukan tuning hyperparameter.