In [17]:
```
import numpy as np
import pandas as pd
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split, GridSearchCV, KFold
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
```

In [18]:
```
digits = load_digits()
X, y = digits.data, digits.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_
```

In [19]:
```
kf = KFold(n_splits=5, shuffle=True, random_state=42)
```

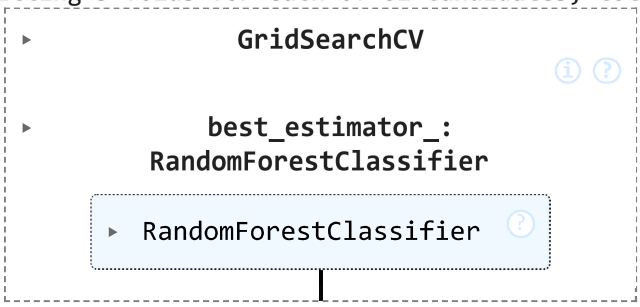
In [20]:
```
rf = RandomForestClassifier(random_state=42)
```

In [21]:
```
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}
grid_search = GridSearchCV(estimator=rf,
                           param_grid=param_grid,
                           cv=kf,
                           scoring='accuracy',
                           n_jobs=-1,
                           verbose=2)
```

In [22]:
```
grid_search.fit(X_train, y_train)
```

```
Fitting 5 folds for each of 81 candidates, totalling 405 fits
```

Out[22]:



In [23]:
```
print("Best Hyperparameters:", grid_search.best_params_)
best_model = grid_search.best_estimator_

y_pred = best_model.predict(X_test)
print("Test Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
Best Hyperparameters: {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_spl
it': 2, 'n_estimators': 200}
Test Accuracy: 0.9694444444444444

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.97      0.98        33
           1       0.93      1.00      0.97        28
           2       1.00      1.00      1.00        33
           3       1.00      0.94      0.97        34
           4       0.98      1.00      0.99        46
           5       0.94      0.96      0.95        47
           6       0.97      0.97      0.97        35
           7       0.97      0.97      0.97        34
           8       0.97      0.93      0.95        30
           9       0.95      0.95      0.95        40

    accuracy                           0.97       360
   macro avg       0.97      0.97      0.97       360
weighted avg       0.97      0.97      0.97       360
```

In [24]:
```python
baseline_model = RandomForestClassifier(random_state=42)
baseline_model.fit(X_train, y_train)
y_base = baseline_model.predict(X_test)
print("Baseline Accuracy:", accuracy_score(y_test, y_base))
improvement = accuracy_score(y_test, y_pred) - accuracy_score(y_test, y_base)
print(f"Accuracy Improvement: {improvement:.4f}")
```

```
Baseline Accuracy: 0.9722222222222222
Accuracy Improvement: -0.0028
```