

STEP 1: Import Required Libraries

```
import numpy as np
import pandas as pd

from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
```

STEP 2: Load Dataset

```
data = load_breast_cancer()

X = data.data
y = data.target
```

STEP 3: Train-Test Split

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.2,
    random_state=42,
    stratify=y
)
```

STEP 4: Train Default Random Forest

```
rf_default = RandomForestClassifier(random_state=42)
rf_default.fit(X_train, y_train)

y_pred_default = rf_default.predict(X_test)

print("Default Model Accuracy:",
      accuracy_score(y_test, y_pred_default))
```

Default Model Accuracy: 0.956140350877193

STEP 5: Define Parameter Grid

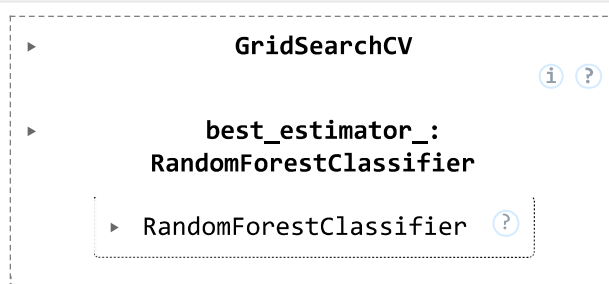
```
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 5, 10, 20],
    'min_samples_split': [2, 5, 10],
```

```
'min_samples_leaf': [1, 2, 4]
}
```

STEP 6: Apply GridSearchCV

```
grid_search = GridSearchCV(
    estimator=RandomForestClassifier(random_state=42),
    param_grid=param_grid,
    cv=5,
    scoring='accuracy',
    n_jobs=-1
)

grid_search.fit(X_train, y_train)
```



STEP 7: Extract Best Model & Parameters

```
print("Best Parameters:", grid_search.best_params_)

best_model = grid_search.best_estimator_

Best Parameters: {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2,
```

STEP 8: Evaluate Tuned Model

```
y_pred_tuned = best_model.predict(X_test)

print("Tuned Model Accuracy:",
      accuracy_score(y_test, y_pred_tuned))

print("\nClassification Report:\n")
print(classification_report(y_test, y_pred_tuned))
```

Tuned Model Accuracy: 0.956140350877193

Classification Report:

| | precision | recall | f1-score | support |
|---|-----------|--------|----------|---------|
| 0 | 0.95 | 0.93 | 0.94 | 42 |

| | | | | |
|--------------|------|------|------|-----|
| 1 | 0.96 | 0.97 | 0.97 | 72 |
| accuracy | | | 0.96 | 114 |
| macro avg | 0.96 | 0.95 | 0.95 | 114 |
| weighted avg | 0.96 | 0.96 | 0.96 | 114 |

STEP 9: Compare Default vs Tuned

```
comparison = pd.DataFrame({
    "Model": ["Default Random Forest", "Tuned Random Forest"],
    "Accuracy": [
        accuracy_score(y_test, y_pred_default),
        accuracy_score(y_test, y_pred_tuned)
    ]
})

print(comparison)
```

| | Model | Accuracy |
|---|-----------------------|----------|
| 0 | Default Random Forest | 0.95614 |
| 1 | Tuned Random Forest | 0.95614 |