**Hyperparameter Tuning for RandomForestRegressor Using GridSearchCV**

**Objective:**

The code performs hyperparameter tuning for a RandomForestRegressor using GridSearchCV. It defines a parameter grid with various combinations of 'n\_estimators', 'max\_features', 'max\_depth', 'min\_samples\_split', and 'min\_samples\_leaf'. The RandomForestRegressor model is instantiated, and GridSearchCV is set up with the defined parameter grid, using 3-fold cross-validation. The grid search is then fitted to the training data (X\_train and y\_train), and the best hyperparameters are identified and printed. This process aims to optimize the model's performance by systematically evaluating different hyperparameter configurations.

**Result:**

**Code1:**

From the above code, I have filtered the results to only include those where the computation time is less than 10 seconds. Among these filtered results, the best hyperparameters are selected to optimize the model's performance.

**Code2:**

The hyperparameter tuning process identified the best configuration for the RandomForestRegressor as follows: **“max\_depth:10, max\_features:sqrt, min\_samples\_leaf:1, min\_samples\_split:2, and n\_estimators:200”.** This configuration achieved a best score of 0.97, indicating a highly effective model performance.

**Code1:**

# Define parameter grid

param\_grid = {

'n\_estimators': [100, 200, 300],

'max\_features': ['sqrt', 'log2'],

'max\_depth': [None, 10, 20, 30],

'min\_samples\_split': [2, 5, 10],

'min\_samples\_leaf': [1, 2, 4]

}

# Instantiate the model

rf\_model = RandomForestRegressor()

# Instantiate Grid Search

grid\_search = GridSearchCV(estimator=rf\_model, param\_grid=param\_grid, cv=3, n\_jobs=1, verbose=2)

# Fit Grid Search

grid\_search.fit(X\_train, y\_train)

# Get best parameters

best\_params = grid\_search.best\_params\_

print(f'Best parameters: {best\_params}')

**Output:**

'max\_depth': None, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 5, 'n\_estimators': 100

'max\_depth': None, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 10, 'n\_estimators': 100

'max\_depth': None, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 5, 'n\_estimators': 100

'max\_depth': None, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 10, 'n\_estimators': 100

'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 100

'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 200

'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 5, 'n\_estimators': 100

'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 10, 'n\_estimators': 100

'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 10, 'n\_estimators': 200

'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 100

'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 200

'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 100

'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 200

'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 10, 'n\_estimators': 100

'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 100

'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 200

'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 5, 'n\_estimators': 100

'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 5, 'n\_estimators': 200

'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 10, 'n\_estimators': 100

'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 10, 'n\_estimators': 200

'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 100

'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 200

'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 1, 'min\_samples\_split': 5, 'n\_estimators': 100

'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 1, 'min\_samples\_split': 5, 'n\_estimators': 200

'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 1, 'min\_samples\_split': 10, 'n\_estimators': 100

'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 1, 'min\_samples\_split': 10, 'n\_estimators': 200

'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 100

'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 200

'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 100

'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 200

'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 2, 'min\_samples\_split': 10, 'n\_estimators': 100

'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 100

'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 200

'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 5, 'n\_estimators': 100

'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 10, 'n\_estimators': 100

'max\_depth': 20, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 10, 'n\_estimators': 100

'max\_depth': 20, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 100

'max\_depth': 20, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 100

'max\_depth': 20, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 10, 'n\_estimators': 100

'max\_depth': 20, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 100

'max\_depth': 20, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 5, 'n\_estimators': 100

'max\_depth': 20, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 10, 'n\_estimators': 100

'max\_depth': 20, 'max\_features': 'log2', 'min\_samples\_leaf': 2, 'min\_samples\_split': 10, 'n\_estimators': 100

'max\_depth': 20, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 100

**Code:**

import numpy as np

import pandas as pd

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import train\_test\_split, cross\_val\_score

from sklearn.metrics import accuracy\_score

# Example dataset (replace with your actual dataset)

# Assuming 'data' is a pandas DataFrame with features and 'target' is the target variable

# data = pd.read\_csv('your\_dataset.csv')

# X = data.drop(columns=['target'])

# y = data['target']

# Sample data (for demonstration purposes)

from sklearn.datasets import load\_iris

iris = load\_iris()

X = iris.data

y = iris.target

# Hyperparameter configurations

configs = [

{'max\_depth': None, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 5, 'n\_estimators': 100},

{'max\_depth': None, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 10, 'n\_estimators': 100},

{'max\_depth': None, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 5, 'n\_estimators': 100},

{'max\_depth': None, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 10, 'n\_estimators': 100},

{'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 100},

{'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 200},

{'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 5, 'n\_estimators': 100},

{'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 10, 'n\_estimators': 100},

{'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 10, 'n\_estimators': 200},

{'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 100},

{'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 200},

{'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 100},

{'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 200},

{'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 10, 'n\_estimators': 100},

{'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 100},

{'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 200},

{'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 5, 'n\_estimators': 100},

{'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 5, 'n\_estimators': 200},

{'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 10, 'n\_estimators': 100},

{'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 10, 'n\_estimators': 200},

{'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 100},

{'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 200},

{'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 1, 'min\_samples\_split': 5, 'n\_estimators': 100},

{'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 1, 'min\_samples\_split': 5, 'n\_estimators': 200},

{'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 1, 'min\_samples\_split': 10, 'n\_estimators': 100},

{'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 1, 'min\_samples\_split': 10, 'n\_estimators': 200},

{'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 100},

{'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 200},

{'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 100},

{'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 200},

{'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 2, 'min\_samples\_split': 10, 'n\_estimators': 100},

{'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 100},

{'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 200},

{'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 5, 'n\_estimators': 100},

{'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 10, 'n\_estimators': 100},

{'max\_depth': 20, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 10, 'n\_estimators': 100},

{'max\_depth': 20, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 100},

{'max\_depth': 20, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 100},

{'max\_depth': 20, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 10, 'n\_estimators': 100},

{'max\_depth': 20, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 100},

{'max\_depth': 20, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 5, 'n\_estimators': 100},

{'max\_depth': 20, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 10, 'n\_estimators': 100},

{'max\_depth': 20, 'max\_features': 'log2', 'min\_samples\_leaf': 2, 'min\_samples\_split': 10, 'n\_estimators': 100},

{'max\_depth': 20, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 100}

]

# Function to evaluate a configuration

def evaluate\_config(config):

model = RandomForestClassifier(\*\*config)

scores = cross\_val\_score(model, X, y, cv=5, scoring='accuracy') # Change scoring to your preferred metric

return np.mean(scores)

# Evaluate all configurations and find the best one

best\_score = 0

best\_config = None

for config in configs:

score = evaluate\_config(config)

print(f"Config: {config}, Score: {score}")

if score > best\_score:

best\_score = score

best\_config = config

print(f"\nBest Config: {best\_config}, Best Score: {best\_score}")

**Output:**

Config: {'max\_depth': None, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 5, 'n\_estimators': 100}, Score: 0.96

Config: {'max\_depth': None, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 10, 'n\_estimators': 100}, Score: 0.96

Config: {'max\_depth': None, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 5, 'n\_estimators': 100}, Score: 0.9533333333333334

Config: {'max\_depth': None, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 10, 'n\_estimators': 100}, Score: 0.9533333333333334

Config: {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 100}, Score: 0.96

Config: {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 200}, Score: 0.9666666666666668

Config: {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 5, 'n\_estimators': 100}, Score: 0.9666666666666668

Config: {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 10, 'n\_estimators': 100}, Score: 0.9533333333333334

Config: {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 10, 'n\_estimators': 200}, Score: 0.9533333333333334

Config: {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 100}, Score: 0.9533333333333334

Config: {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 200}, Score: 0.9533333333333334

Config: {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 100}, Score: 0.9533333333333334

Config: {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 200}, Score: 0.9533333333333334

Config: {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 10, 'n\_estimators': 100}, Score: 0.9533333333333334

Config: {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 100}, Score: 0.9533333333333334

Config: {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 200}, Score: 0.9533333333333334

Config: {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 5, 'n\_estimators': 100}, Score: 0.9533333333333334

Config: {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 5, 'n\_estimators': 200}, Score: 0.9533333333333334

Config: {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 10, 'n\_estimators': 100}, Score: 0.96

Config: {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 10, 'n\_estimators': 200}, Score: 0.9533333333333334

Config: {'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 100}, Score: 0.9666666666666668

Config: {'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 200}, Score: 0.96

Config: {'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 1, 'min\_samples\_split': 5, 'n\_estimators': 100}, Score: 0.9533333333333334

Config: {'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 1, 'min\_samples\_split': 5, 'n\_estimators': 200}, Score: 0.9666666666666668

Config: {'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 1, 'min\_samples\_split': 10, 'n\_estimators': 100}, Score: 0.9666666666666668

Config: {'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 1, 'min\_samples\_split': 10, 'n\_estimators': 200}, Score: 0.9666666666666668

Config: {'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 100}, Score: 0.9533333333333334

Config: {'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 200}, Score: 0.9533333333333334

Config: {'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 100}, Score: 0.9666666666666668

Config: {'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 200}, Score: 0.9533333333333334

Config: {'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 2, 'min\_samples\_split': 10, 'n\_estimators': 100}, Score: 0.9533333333333334

Config: {'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 100}, Score: 0.9533333333333334

Config: {'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 200}, Score: 0.9666666666666668

Config: {'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 5, 'n\_estimators': 100}, Score: 0.9666666666666668

Config: {'max\_depth': 10, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 10, 'n\_estimators': 100}, Score: 0.9533333333333334

Config: {'max\_depth': 20, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 10, 'n\_estimators': 100}, Score: 0.9666666666666668

Config: {'max\_depth': 20, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 100}, Score: 0.9533333333333334

Config: {'max\_depth': 20, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 100}, Score: 0.96

Config: {'max\_depth': 20, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 10, 'n\_estimators': 100}, Score: 0.9533333333333334

Config: {'max\_depth': 20, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 100}, Score: 0.9533333333333334

Config: {'max\_depth': 20, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 5, 'n\_estimators': 100}, Score: 0.9533333333333334

Config: {'max\_depth': 20, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 10, 'n\_estimators': 100}, Score: 0.9533333333333334

Config: {'max\_depth': 20, 'max\_features': 'log2', 'min\_samples\_leaf': 2, 'min\_samples\_split': 10, 'n\_estimators': 100}, Score: 0.9533333333333334

Config: {'max\_depth': 20, 'max\_features': 'log2', 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 100}, Score: 0.9533333333333334

Best Config: {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 200}, Best Score: 0.9666666666666668