

213537582?Snlc, > n C / - 2

try 1<sup>o</sup>

decision tree

```
from sklearn.tree import DecisionTreeClassifier
|
# model_decision_tree = DecisionTreeClassifier(...)
# model_decision_tree.fit(...)
model_decision_tree = DecisionTreeClassifier(max_depth=4, random_state=42)
model_decision_tree.fit(X_train, y_train)

print("Decision Tree trained successfully!")
```

Random Forests

```
from sklearn.ensemble import RandomForestClassifier
|
# model_random_forest = RandomForestClassifier(...)
# model_random_forest.fit(...)
model_random_forest = RandomForestClassifier(n_estimators=100, max_depth=4, random_state=42)
model_random_forest.fit(X_train, y_train)

print("Random Forest trained successfully!")
```

SLIDES action towards linear

8 decision tree

Ada Boost

```
from sklearn.ensemble import AdaBoostClassifier
|
# model_adaboost = AdaBoostClassifier(...)
# model_adaboost.fit(...)
model_adaboost = AdaBoostClassifier(n_estimators=50, random_state=42)
model_adaboost.fit(X_train, y_train)

print("AdaBoost trained successfully!")
©
```

(2)

Random Forest Performance:

Accuracy: 0.9708

Precision: 0.9640

Recall: 0.9907

F1 Score: 0.9772

I

(1)

Decision Tree Performance:

Accuracy: 0.9532

Precision: 0.9630

Recall: 0.9630

F1 Score: 0.9630

AdaBoost Performance:

Accuracy: 0.9766

Precision: 0.9815

Recall: 0.9815

F1 Score: 0.9815

The best model is: AdaBoost with F1 Score = 0.9815

I

(3)

Try 23

```
[10] from sklearn.tree import DecisionTreeClassifier  
  
# model_decision_tree = DecisionTreeClassifier(...)  
# model_decision_tree.fit(...)  
model_decision_tree = DecisionTreeClassifier(max_depth=20, random_state=40)  
model_decision_tree.fit(X_train, y_train)  
  
print("Decision Tree trained successfully!")
```

```
from sklearn.ensemble import AdaBoostClassifier  
  
# model_adaboost = AdaBoostClassifier(...)  
# model_adaboost.fit(...)  
model_adaboost = AdaBoostClassifier(n_estimators=50, random_state=42)  
model_adaboost.fit(X_train, y_train)  
  
print("AdaBoost trained successfully!")  
①
```

```
from sklearn.ensemble import RandomForestClassifier  
  
# model_random_forest = RandomForestClassifier(...)  
# model_random_forest.fit(...)  
model_random_forest = RandomForestClassifier(n_estimators=100, max_depth=4, random_state=42)  
model_random_forest.fit(X_train, y_train)  
  
print("Random Forest trained successfully!")
```

Decision Tree Performance:  
Accuracy: 0.9298  
Precision: 0.9615  
Recall: 0.9259  
F1 Score: 0.9434

Random Forest Performance:  
Accuracy: 0.9708  
Precision: 0.9640  
Recall: 0.9907  
F1 Score: 0.9772

AdaBoost Performance:  
Accuracy: 0.9766  
Precision: 0.9815  
Recall: 0.9815  
F1 Score: 0.9815

The best model is: AdaBoost with F1 Score = 0.9815

# Try 3/8

```
▶ from sklearn.tree import DecisionTreeClassifier  
  
# model_decision_tree = DecisionTreeClassifier(...)  
# model_decision_tree.fit(...)  
model_decision_tree = DecisionTreeClassifier(max_depth=5, random_state=2)  
model_decision_tree.fit(X_train, y_train)  
I  
print("Decision Tree trained successfully!")
```

```
from sklearn.ensemble import RandomForestClassifier  
  
# model_random_forest = RandomForestClassifier(...)  
# model_random_forest.fit(...)  
model_random_forest = RandomForestClassifier(n_estimators=100, max_depth=4, random_state=42)  
model_random_forest.fit(X_train, y_train)  
  
print("Random Forest trained successfully!")
```

```
from sklearn.ensemble import AdaBoostClassifier  
  
# model_adaboost = AdaBoostClassifier(...)  
# model_adaboost.fit(...)  
model_adaboost = AdaBoostClassifier(n_estimators=50, random_state=42)  
model_adaboost.fit(X_train, y_train)  
  
print("AdaBoost trained successfully!")  
①
```

```
→ Decision Tree Performance:  
Accuracy: 0.9474  
Precision: 0.9541  
Recall: 0.9630  
F1 Score: 0.9585  
  
Random Forest Performance:  
Accuracy: 0.9708  
Precision: 0.9640  
Recall: 0.9907  
F1 Score: 0.9772  
  
AdaBoost Performance:  
Accuracy: 0.9766  
Precision: 0.9815  
Recall: 0.9815  
F1 Score: 0.9815  
  
The best model is: AdaBoost with F1 Score = 0.9815
```

Try 78

```
28] from sklearn.tree import DecisionTreeClassifier  
  
    # model_decision_tree = DecisionTreeClassifier(...)  
    # model_decision_tree.fit(...)  
    model_decision_tree = DecisionTreeClassifier(max_depth=51, random_state=62)  
    model_decision_tree.fit(X_train, y_train)  
  
    print("Decision Tree trained successfully!")
```

```
from sklearn.ensemble import AdaBoostClassifier  
  
# model_adaboost = AdaBoostClassifier(...)  
# model_adaboost.fit(...)  
model_adaboost = AdaBoostClassifier(n_estimators=50, random_state=42)  
model_adaboost.fit(X_train, y_train)  
  
print("AdaBoost trained successfully!")  
①
```

```
from sklearn.ensemble import RandomForestClassifier  
  
# model_random_forest = RandomForestClassifier(...)  
# model_random_forest.fit(...)  
model_random_forest = RandomForestClassifier(n_estimators=100, max_depth=4, random_state=42)  
model_random_forest.fit(X_train, y_train)  
  
print("Random Forest trained successfully!")
```

→ Decision Tree Performance:  
Accuracy: 0.9240  
Precision: 0.9524  
Recall: 0.9259  
F1 Score: 0.9390

Random Forest Performance:  
Accuracy: 0.9708  
Precision: 0.9640  
Recall: 0.9907  
F1 Score: 0.9772

AdaBoost Performance:  
Accuracy: 0.9766  
Precision: 0.9815  
Recall: 0.9815  
F1 Score: 0.9815

The best model is: AdaBoost with F1 Score = 0.9815

Try So

```
[32] from sklearn.tree import DecisionTreeClassifier  
| # model_decision_tree = DecisionTreeClassifier(...)  
| # model_decision_tree.fit(...)  
| model_decision_tree = DecisionTreeClassifier(max_depth=20, random_state=42)  
| model_decision_tree.fit(X_train, y_train)  
|  
| print("Decision Tree trained successfully!")
```

```
from sklearn.ensemble import AdaBoostClassifier  
  
# model_adaboost = AdaBoostClassifier(...)  
# model_adaboost.fit(...)  
model_adaboost = AdaBoostClassifier(n_estimators=50, random_state=42)  
model_adaboost.fit(X_train, y_train)  
  
print("AdaBoost trained successfully!")
```

```
from sklearn.ensemble import RandomForestClassifier  
  
# model_random_forest = RandomForestClassifier(...)  
# model_random_forest.fit(...)  
model_random_forest = RandomForestClassifier(n_estimators=100, max_depth=4, random_state=42)  
model_random_forest.fit(X_train, y_train)  
  
print("Random Forest trained successfully!")
```

#### → Decision Tree Performance:

Accuracy: 0.9357

Precision: 0.9709

Recall: 0.9259

F1 Score: 0.9479

#### Random Forest Performance:

Accuracy: 0.9708

Precision: 0.9640

Recall: 0.9907

F1 Score: 0.9772

#### AdaBoost Performance:

Accuracy: 0.9766

Precision: 0.9815

Recall: 0.9815

F1 Score: 0.9815

The best model is: AdaBoost with F1 Score = 0.9815

# Random forests

Try 1 8

```
= RandomForestClassifier(...)  
fit(...)  
RandomForestClassifier(n_estimators=100, max_depth=44, random_state=40)  
t(X_train, y_train)  
  
trained successfully!"
```

```
from sklearn.ensemble import AdaBoostClassifier  
  
# model_adaboost = AdaBoostClassifier(...)  
# model_adaboost.fit(...)  
model_adaboost = AdaBoostClassifier(n_estimators=50, random_state=42)  
model_adaboost.fit(X_train, y_train)  
  
print("AdaBoost trained successfully!")  
()
```

```
[32] from sklearn.tree import DecisionTreeClassifier  
  
# model_decision_tree = DecisionTreeClassifier(...)  
# model_decision_tree.fit(...)  
model_decision_tree = DecisionTreeClassifier(max_depth=20, random_state=20)  
model_decision_tree.fit(X_train, y_train)  
  
print("Decision Tree trained successfully!")
```

## Decision Tree Performance:

Accuracy: 0.9357

Precision: 0.9709

Recall: 0.9259

F1 Score: 0.9479

## Random Forest Performance:

Accuracy: 0.9708

Precision: 0.9725

Recall: 0.9815

F1 Score: 0.9770

## AdaBoost Performance:

Accuracy: 0.9766

Precision: 0.9815

Recall: 0.9815

F1 Score: 0.9815

The best model is: AdaBoost with F1 Score = 0.9815

Try 28

```
[36] ensemble import RandomForestClassifier  
  
m_forest = RandomForestClassifier(...)  
m_forest.fit(...)  
forest = RandomForestClassifier(n_estimators=100, max_depth=4, random_state=12)  
forest.fit(X_train, y_train)  
  
Forest trained successfully!"
```

```
from sklearn.ensemble import AdaBoostClassifier  
  
# model_adaboost = AdaBoostClassifier(...)  
# model_adaboost.fit(...)  
model_adaboost = AdaBoostClassifier(n_estimators=50, random_state=42)  
model_adaboost.fit(X_train, y_train)  
  
print("AdaBoost trained successfully!")
```

```
[32] from sklearn.tree import DecisionTreeClassifier  
  
# model_decision_tree = DecisionTreeClassifier(...)  
# model_decision_tree.fit(...)  
model_decision_tree = DecisionTreeClassifier(max_depth=20, random_state=20)  
model_decision_tree.fit(X_train, y_train)  
  
print("Decision Tree trained successfully!")
```

```
→ Decision Tree Performance:  
Accuracy: 0.9357  
Precision: 0.9709  
Recall: 0.9259  
F1 Score: 0.9479  
  
Random Forest Performance:  
Accuracy: 0.9649  
Precision: 0.9636  
Recall: 0.9815  
F1 Score: 0.9725  
  
AdaBoost Performance:  
Accuracy: 0.9766  
Precision: 0.9815  
Recall: 0.9815  
F1 Score: 0.9815  
  
The best model is: AdaBoost with F1 Score = 0.9815
```

Try 38

```
[1]: ensemble import RandomForestClassifier  
  
[2]: m_forest = RandomForestClassifier(...)  
m_forest.fit(...)  
forest = RandomForestClassifier(n_estimators=100, max_depth=10, random_state=2)  
forest.fit(X_train, y_train)  
  
Forest trained successfully!
```

Random Forest trained successfully!

```
[32]: from sklearn.tree import DecisionTreeClassifier  
  
# model_decision_tree = DecisionTreeClassifier(...)  
# model_decision_tree.fit(...)  
model_decision_tree = DecisionTreeClassifier(max_depth=20, random_state=20)  
model_decision_tree.fit(X_train, y_train)  
  
print("Decision Tree trained successfully!")
```

```
from sklearn.ensemble import AdaBoostClassifier  
  
# model_adaboost = AdaBoostClassifier(...)  
# model_adaboost.fit(...)  
model_adaboost = AdaBoostClassifier(n_estimators=50, random_state=42)  
model_adaboost.fit(X_train, y_train)  
  
print("AdaBoost trained successfully!")
```

①

#### → Decision Tree Performance:

Accuracy: 0.9357

Precision: 0.9709

Recall: 0.9259

F1 Score: 0.9479

#### Random Forest Performance:

Accuracy: 0.9649

Precision: 0.9636

Recall: 0.9815

F1 Score: 0.9725

#### AdaBoost Performance:

Accuracy: 0.9766

Precision: 0.9815

Recall: 0.9815

F1 Score: 0.9815

The best model is: AdaBoost with F1 Score = 0.9815

Try vs

```
[40] ensemble import RandomForestClassifier  
  
    m_forest = RandomForestClassifier(...)  
    m_forest.fit(...)  
    forest = RandomForestClassifier(n_estimators=100, max_depth=14, random_state=42)  
    forest.fit(X_train, y_train)  
  
    Forest trained successfully!"
```

```
[32] from sklearn.tree import DecisionTreeClassifier  
  
    # model_decision_tree = DecisionTreeClassifier(...)  
    # model_decision_tree.fit(...)  
    model_decision_tree = DecisionTreeClassifier(max_depth=20, random_state=20)  
    model_decision_tree.fit(X_train, y_train)  
  
    print("Decision Tree trained successfully!")
```

```
from sklearn.ensemble import AdaBoostClassifier  
  
# model_adaboost = AdaBoostClassifier(...)  
# model_adaboost.fit(...)  
model_adaboost = AdaBoostClassifier(n_estimators=50, random_state=42)  
model_adaboost.fit(X_train, y_train)  
  
print("AdaBoost trained successfully!")
```

#### → Decision Tree Performance:

Accuracy: 0.9357

Precision: 0.9709

Recall: 0.9259

F1 Score: 0.9479

#### Random Forest Performance:

Accuracy: 0.9708

Precision: 0.9640

Recall: 0.9907

F1 Score: 0.9772

#### AdaBoost Performance:

Accuracy: 0.9766

Precision: 0.9815

Recall: 0.9815

F1 Score: 0.9815

The best model is: AdaBoost with F1 Score = 0.9815

Try 58

```
ensemble import RandomForestClassifier

m_forest = RandomForestClassifier(...)
m_forest.fit(...)
forest = RandomForestClassifier(n_estimators=100, max_depth=64, random_state=22)
forest.fit(x_train, y_train)

Forest trained successfully!"
```

```
[32] from sklearn.tree import DecisionTreeClassifier
      # model_decision_tree = DecisionTreeClassifier(...)
      # model_decision_tree.fit...
model_decision_tree = DecisionTreeClassifier(max_depth=20, random_state=20)
model_decision_tree.fit(x_train, y_train)

print("Decision Tree trained successfully!")
```

```
from sklearn.ensemble import AdaBoostClassifier

# model_adaboost = AdaBoostClassifier(...)
# model_adaboost.fit...
model_adaboost = AdaBoostClassifier(n_estimators=50, random_state=42)
model_adaboost.fit(x_train, y_train)

print("AdaBoost trained successfully!")
()
```

#### Decision Tree Performance:

Accuracy: 0.9357  
Precision: 0.9709  
Recall: 0.9259  
F1 Score: 0.9479

#### Random Forest Performance:

Accuracy: 0.9708  
Precision: 0.9640  
Recall: 0.9907  
F1 Score: 0.9772

#### AdaBoost Performance:

Accuracy: 0.9766  
Precision: 0.9815  
Recall: 0.9815  
F1 Score: 0.9815

The best model is: AdaBoost with F1 Score = 0.9815

# AdaBoost vs Tree

```
[44] from sklearn.ensemble import AdaBoostClassifier
```

```
# model_adaboost = AdaBoostClassifier(...)  
# model_adaboost.fit(...)  
model_adaboost = AdaBoostClassifier(n_estimators=5, random_state=4)  
model_adaboost.fit(X_train, y_train)  
  
print("AdaBoost trained successfully!")
```

```
[32] from sklearn.tree import DecisionTreeClassifier  
# model_decision_tree = DecisionTreeClassifier(...)  
# model_decision_tree.fit(...)  
model_decision_tree = DecisionTreeClassifier(max_depth=20, random_state=20)  
model_decision_tree.fit(X_train, y_train)  
print("Decision Tree trained successfully!")
```

```
● ensemble import RandomForestClassifier  
  
m_forest = RandomForestClassifier(...)  
m_forest.fit(...)  
forest = RandomForestClassifier(n_estimators=100, max_depth=64, random_state=22)  
forest.fit(X_train, y_train)  
  
Forest trained successfully!)
```

## Decision Tree Performance:

Accuracy: 0.9357

Precision: 0.9709

Recall: 0.9259

F1 Score: 0.9479

## Random Forest Performance:

Accuracy: 0.9708

Precision: 0.9640

Recall: 0.9907

F1 Score: 0.9772

## AdaBoost Performance:

Accuracy: 0.9591

Precision: 0.9633

Recall: 0.9722

F1 Score: 0.9677

The best model is: Random Forest with F1 Score = 0.9772

Try 2o

```
▶ from sklearn.ensemble import AdaBoostClassifier  
  
# model_adaboost = AdaBoostClassifier(...)  
# model_adaboost.fit(...)  
model_adaboost = AdaBoostClassifier(n_estimators=15, random_state=24)  
model_adaboost.fit(X_train, y_train)  
  
print("AdaBoost trained successfully!")
```

```
✓ [32] from sklearn.tree import DecisionTreeClassifier  
  
# model_decision_tree = DecisionTreeClassifier(...)  
# model_decision_tree.fit(...)  
model_decision_tree = DecisionTreeClassifier(max_depth=20, random_state=20)  
model_decision_tree.fit(X_train, y_train)  
  
print("Decision Tree trained successfully!")
```

```
▶ ensemble import RandomForestClassifier  
  
m_forest = RandomForestClassifier(...)  
m_forest.fit(...)  
forest = RandomForestClassifier(n_estimators=100, max_depth=64, random_state=22)  
forest.fit(X_train, y_train)  
  
Forest trained successfully!
```

#### Decision Tree Performance:

Accuracy: 0.9357

Precision: 0.9709

Recall: 0.9259

F1 Score: 0.9479

#### Random Forest Performance:

Accuracy: 0.9708

Precision: 0.9640

Recall: 0.9907

F1 Score: 0.9772

#### AdaBoost Performance:

Accuracy: 0.9708

Precision: 0.9813

Recall: 0.9722

F1 Score: 0.9767

The best model is: Random Forest with F1 Score = 0.9772

Try 3B

```
from sklearn.ensemble import AdaBoostClassifier  
  
# model_adaboost = AdaBoostClassifier(...)  
# model_adaboost.fit(...)  
model_adaboost = AdaBoostClassifier(n_estimators=45, random_state=34)  
model_adaboost.fit(X_train, y_train)  
  
print("AdaBoost trained successfully!")
```

```
[32] from sklearn.tree import DecisionTreeClassifier  
  
# model_decision_tree = DecisionTreeClassifier(...)  
# model_decision_tree.fit(...)  
model_decision_tree = DecisionTreeClassifier(max_depth=20, random_state=20)  
model_decision_tree.fit(X_train, y_train)  
  
print("Decision Tree trained successfully!")
```

```
ensemble import RandomForestClassifier  
  
# forest = RandomForestClassifier(...)  
# forest.fit(...)  
forest = RandomForestClassifier(n_estimators=100, max_depth=64, random_state=22)  
forest.fit(X_train, y_train)  
  
print("Forest trained successfully!")
```

→ **Decision Tree Performance:**  
**Accuracy: 0.9357**  
**Precision: 0.9709**  
**Recall: 0.9259**  
**F1 Score: 0.9479**

**Random Forest Performance:**  
**Accuracy: 0.9703**  
**Precision: 0.9640**  
**Recall: 0.9907**  
**F1 Score: 0.9772**

**AdaBoost Performance:**  
**Accuracy: 0.9766**  
**Precision: 0.9815**  
**Recall: 0.9815**  
**F1 Score: 0.9815**

The best model is: AdaBoost with F1 Score = 0.9815

Try 2

```
▶ from sklearn.ensemble import AdaBoostClassifier  
    # model_adaboost = AdaBoostClassifier(...)  
    # model_adaboost.fit(...)  
model_adaboost = AdaBoostClassifier(n_estimators=10, random_state=34)  
model_adaboost.fit(X_train, y_train)  
print("AdaBoost trained successfully!")
```

```
✓ [32] from sklearn.tree import DecisionTreeClassifier  
    # model_decision_tree = DecisionTreeClassifier(...)  
    # model_decision_tree.fit(...)  
model_decision_tree = DecisionTreeClassifier(max_depth=20, random_state=20)  
model_decision_tree.fit(X_train, y_train)  
print("Decision Tree trained successfully!")
```

```
▶ ensemble import RandomForestClassifier  
    m_forest = RandomForestClassifier(...)  
    m_forest.fit(...)  
forest = RandomForestClassifier(n_estimators=100, max_depth=64, random_state=22)  
forest.fit(X_train, y_train)  
print("Forest trained successfully!")
```

#### → Decision Tree Performance:

Accuracy: 0.9357

Precision: 0.9709

Recall: 0.9259

F1 Score: 0.9479

#### Random Forest Performance:

Accuracy: 0.9708

Precision: 0.9640

Recall: 0.9907

F1 Score: 0.9772

#### AdaBoost Performance:

Accuracy: 0.9708

Precision: 0.9725

Recall: 0.9815

F1 Score: 0.9770

The best model is: Random Forest with F1 Score = 0.9772

Try S8

```
from sklearn.ensemble import AdaBoostClassifier  
  
# model_adaboost = AdaBoostClassifier(...)  
# model_adaboost.fit(...)  
model_adaboost = AdaBoostClassifier(n_estimators=55, random_state=4)  
model_adaboost.fit(X_train, y_train)  
  
print("AdaBoost trained successfully!")
```

```
[32] from sklearn.tree import DecisionTreeClassifier  
  
# model_decision_tree = DecisionTreeClassifier(...)  
# model_decision_tree.fit(...)  
model_decision_tree = DecisionTreeClassifier(max_depth=20, random_state=20)  
model_decision_tree.fit(X_train, y_train)  
  
print("Decision Tree trained successfully!")
```

```
ensemble import RandomForestClassifier  
  
m_forest = RandomForestClassifier(...)  
m_forest.fit(...)  
forest = RandomForestClassifier(n_estimators=100, max_depth=64, random_state=22)  
forest.fit(X_train, y_train)  
  
Forest trained successfully!"
```

Decision Tree Performance:  
Accuracy: 0.9357  
Precision: 0.9709  
Recall: 0.9259  
F1 Score: 0.9479

Random Forest Performance:  
Accuracy: 0.9708  
Precision: 0.9640  
Recall: 0.9907  
F1 Score: 0.9772

AdaBoost Performance:  
Accuracy: 0.9825  
Precision: 0.9907  
Recall: 0.9815  
F1 Score: 0.9860

The best model is: AdaBoost with F1 Score = 0.9860

10/10  
1/1  
2/2