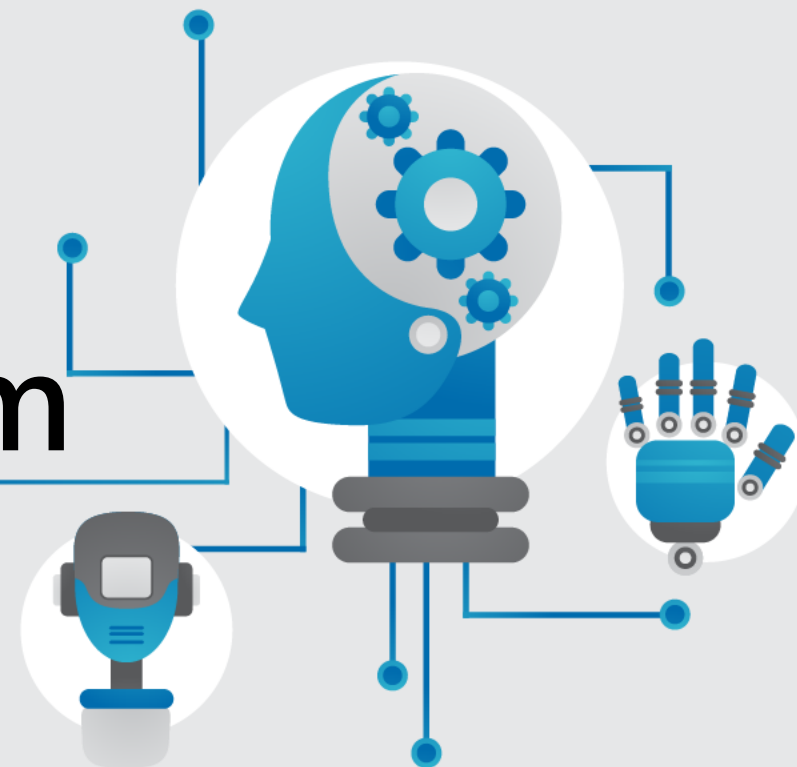


Scikit-learn Random Forest Classifier






Random Forest Classifier介紹

機器學習實務



 [Install](#) [User Guide](#) [API](#) [Examples](#) [More ▾](#)

[Prev](#) [Up](#) [Next](#)

scikit-learn 0.22.2
[Other versions](#)

Please [cite us](#) if you use the software.

3.2.4.3.1.
sklearn.ensemble.RandomForestClassifier

3.2.4.3.1.1. Examples using
sklearn.ensemble.RandomForestClassifier

3.2.4.3.1. sklearn.ensemble.RandomForestClassifier

```
class sklearn.ensemble.RandomForestClassifier(n_estimators=100, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None, ccp_alpha=0.0, max_samples=None)
```

[source]

A random forest classifier.

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement if `bootstrap=True` (default).

Read more in the [User Guide](#).

➤ 隨機森林 (Random Forest)

通過在數據集各個子樣本上訓練大量的決策樹，並且使用平均數來提高預測準確性以及控制過度學習。

➤ sklearn.ensemble.RandomForestClassifier 為Random Forest演算法的實作



Random Forest Classifier 參數說明

機器學習實務

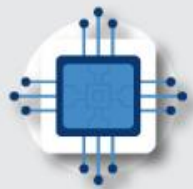


› `class sklearn.ensemble.RandomForestClassifier(n_estimators=100, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None, ccp_alpha=0.0, max_samples=None)`

› Random Forest Classifier 類別常用參數

- `n_estimators`
- `criterion`
- `max_depth`
- `min_samples_split`
- `max_features`
- `bootstrap`



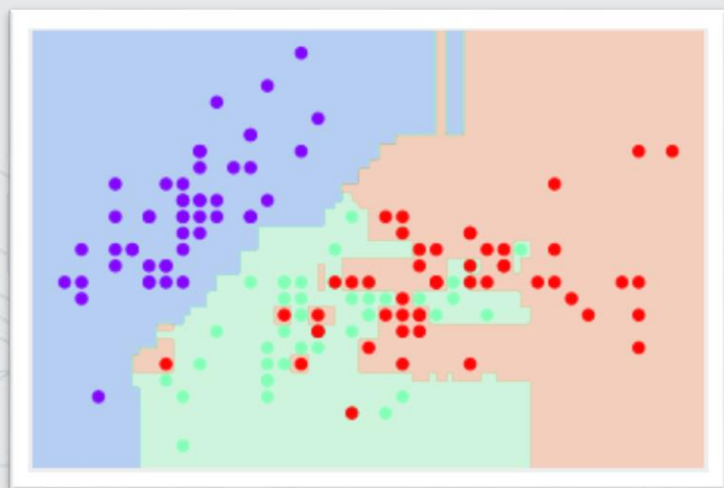


參數 `n_estimators`

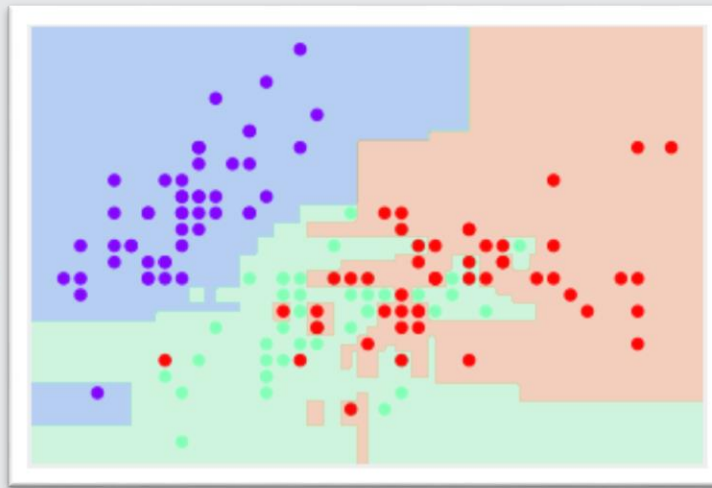


› `n_estimators` : integer, optional (default=100)

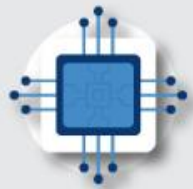
- ✓ 森林中建立子樹的數量，較多的子樹可以讓模型有較好的性能，但是同時也會增加計算量。



`n_estimators=100`



`n_estimators=10`

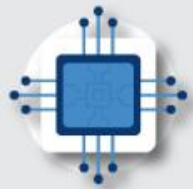


參數 criterion



› **criterion : string, optional (default= “gini”)**

- ✓ 即CART樹做劃分時對特徵的評價標準
- ✓ 預設是基尼係數gini，另一個可選擇的標準是信息增益entropy，是用來選擇節點的最優特徵和切分點的兩個準則。

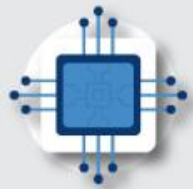


參數 `min_samples_split`



› `min_samples_split` : int, float, optional (default=2)

- ✓ 拆分內部節點所需的最少樣本數。
- ✓ 如果為int，則將`min_samples_split`視為最小值。
- ✓ 如果為float，則`min_samples_split`是一個分數，而 $\text{ceil}(\text{min_samples_split} * n_samples)$ 是每個拆分的最小樣本數。



參數 `max_depth`



› `max_depth` : integer or None, optional (default=None)

- ✓ 樹的最大深度
- ✓ 如果為None，在建立子樹的時候
不會限制子樹的深度，或者直到所有葉子都包含少於
`min_samples_split`個樣本。

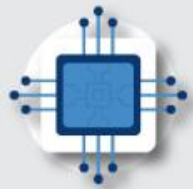


參數 max_features



› max_features : int, float, string or None, optional (default= “auto”)

- ✓ 尋找最佳分割時要考慮的特徵數量
- ✓ 如果為int，則在每個拆分中考慮max_features功能
- ✓ 如果為float，則max_features是一個分數，
並且在每個分割處均考慮 $\text{int}(\text{max_features} * \text{n_features})$ 個特徵
- ✓ 如果為“auto”，則 $\text{max_features} = \text{sqrt}(\text{n_features})$
- ✓ 如果為“sqrt”，則 $\text{max_features} = \text{sqrt}(\text{n_features})$
(與“auto”相同)
- ✓ 如果為“log2”，則 $\text{max_features} = \text{log2}(\text{n_features})$
- ✓ 如果為None，則 $\text{max_features} = \text{n_features}$



參數 bootstrap



› **bootstrap** : boolean, optional (default=True)

- ✓ 建立樹木時是否使用bootstrap樣本
- ✓ 如果為**True**，則將隨機樣本的子集用於構建每個棵樹
- ✓ 如果為**False**，則將整個數據集用於構建每棵樹



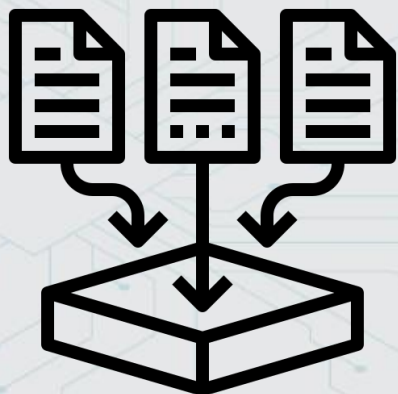
Random Forest Classifier 函式說明

機器學習實務



› Random Forest Classifier 常用函式

- fit
- predict
- score





訓練 (fit)



› 指令 `fit(self, x, y, sample_weight=None)`

› 參數

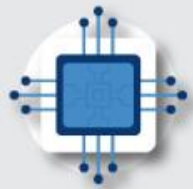
- x : 訓練輸入樣本
- y : 目標值 (分類中的類標籤)

› 回傳 : 訓練後的random forest 物件.

› 說明 : 根據訓練集 (x , y) 建立一個森林樹木

› 範例程式

```
from sklearn.ensemble import RandomForestClassifier  
randomForest = RandomForestClassifier(n_estimators=100)  
randomForest.fit(X_train, y_train)
```



預測 (predict)



› 指令 `predict(self, x)`

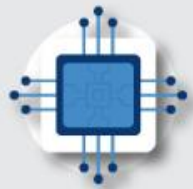
› 參數

- `x` : 輸入樣本

› 回傳 : 預測的類別

› 範例程式

```
from sklearn.ensemble import RandomForestClassifier  
randomForest = RandomForestClassifier(n_estimators=100)  
randomForest.fit(x_train, y_train)  
predictions = randomForest.predict(x_test)
```



評分 (score)



› 指令 `score(self, x, y, sample_weight=None)`

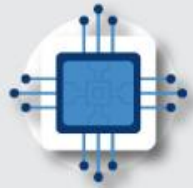
› 參數

- x : 測試樣本
- y : 測試樣本的正确答案

› 回傳：測試樣本的平均準確度

› 範例程式

```
from sklearn.ensemble import RandomForestClassifier  
randomForest = RandomForestClassifier(n_estimators=100)  
randomForest.fit(x_train, y_train)  
accuracy = randomForest.score(x_test, y_test)
```



程式範例 (IRIS)



› 程式碼

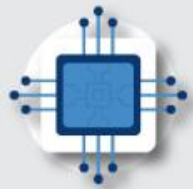
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier
from sklearn import datasets

# 載入資料
iris = datasets.load_iris()
X = iris.data[:, :2] # 只取前兩種特徵
Y = iris.target

# 建立 Random Forest Classifier
randomForest = RandomForestClassifier(n_estimators=100)

# 進行訓練
randomForest.fit(X, Y)

# 繪製座標軸
x_min, x_max = X[:, 0].min() - .5, X[:, 0].max() + .5
y_min, y_max = X[:, 1].min() - .5, X[:, 1].max() + .5
h = .02 # 單位間隔
xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))
```

程式範例 (IRIS)



› 程式碼

```
# 進行預測
Z = randomForest.predict(np.c_[xx.ravel(), yy.ravel()])

# 繪製預測結果
Z = Z.reshape(xx.shape)
plt.figure(1, figsize=(4, 3))
plt.pcolormesh(xx, yy, Z, cmap=plt.cm.Paired)

plt.scatter(X[:, 0], X[:, 1], c=Y, edgecolors='k', cmap=plt.cm.Paired)
plt.xlabel( 'Sepal length' )
plt.ylabel( 'Sepal width' )

plt.xlim(xx.min(), xx.max())
plt.ylim(yy.min(), yy.max())
plt.xticks(())
plt.yticks(())

plt.show()
```



程式範例 (IRIS)

機器學習實務



› 輸出結果

