

D46 : tree based model - 梯度提升機程式碼撰寫



簡報閱讀



範例與作業



問題討論

梯度提升機 - 程式碼撰寫 Coding 練習日



梯度提升機 - 程式碼撰寫 Coding 練習日 >

知識地圖 >

本日知識點目標 >

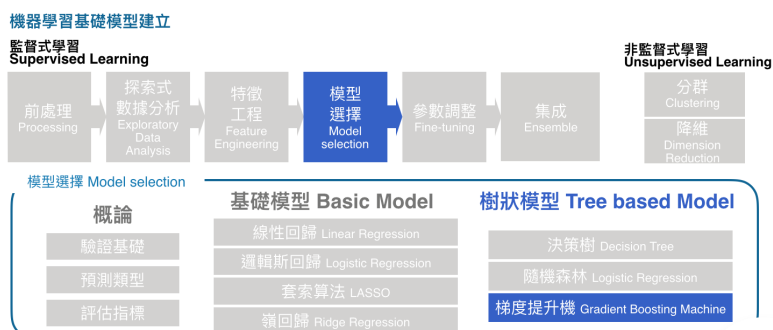
使用 Sklearn 中的梯度提升機 >

使用 Sklearn 中的梯度提升機 >

常見問題 >

延伸閱讀 >

知識地圖



- 了解梯度提升機的程序碼應用
- 如何使用 Sklearn 來建立梯度提升機的模型
- 了解模型中各項參數的意義

使用 Sklearn 中的梯度提升機

- 可以看到如同隨機森林，我們一樣從 `sklearn.ensemble` 這裏 import 進來，代表梯度提升機同樣是個**集成**模型，透過多棵決策樹依序生成來得到結果，緩解原本決策樹容易過擬和的問題，實務上的結果通常也會比決策樹來得好

```
from sklearn.ensemble import
GradientBoostingClassifier
from sklearn.ensemble import
GradientBoostingRegressor
clf = GradientBoostingClassifier()
```

使用 Sklearn 中的梯度提升機

- 同樣是樹的模型，所以像是 `max_depth`, `min_samples_split` 都與決策樹相同
- 可決定要生成數的數量，越多越不容易過擬和，但是運算時間會變長

```
from sklearn.ensemble import
GradientBoostingClassifier
```

```
clf = GradientBoostingClassifier()
```

念相同但實作稍微不同

learning_rate=0.1, #每棵樹對最終結果的影響，
應與 n_estimators 成反比

n_estimators=100 #決策樹的數量

)

常見問題

Q：隨機森林與梯度提升機的特徵重要性結果不相同？

A：決策樹計算特徵重要性的概念是，觀察某一特徵被用來切分的次數而定。假設有兩個一模一樣的特徵，在隨機森林中每棵樹皆為獨立，因此兩個特徵皆有可能被使用，最終統計出來的次數會被均分。在梯度提升機中，每棵樹皆有關連，因此模型僅會使用其中一個特徵，另一個相同特徵的重要性則會消失

[參考資料](#)

延伸閱讀

- 完整調參數攻略-如何使用 Python 調整梯度提升機的超參數 (英文)
 - [complete-guide-parameter-tuning-gradient-boosting-gbm](#)

解題時間

[下一步：閱讀範例與完成作業](#)

