Day 21

XGBoost (迴歸器)

[全民瘋AI系列]

第12屆 iT邦幫忙 鐵人賽

Day 21 學習目標

01 **了解 XGBoost Regression**

Boosting vs Decision tree & Bagging vs. Boosting

02 **實作 XGBoost 迴歸器**

查看 XGBoost 在簡單線性迴歸和非線性回歸表現

Part 1

XGBoost (迴歸器)

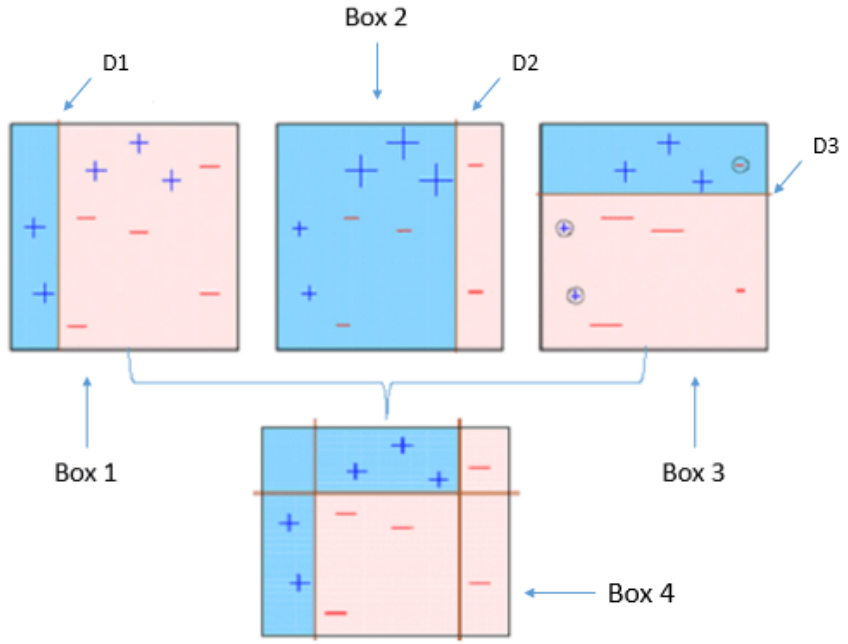
觀念講解

第12屆 iT邦幫忙 鐵人賽

XGBoost Regression

XGBoost 是許多資料科學競賽中常出現的常勝軍。也是當今熱門的ML演算法。

- 使用許多策略去防止過擬合
- early stop 機制可以提早結束訓練
- 可手動設置樣本權重以及學習速率



Boosting vs Decision tree

- 決策樹通常為一棵複雜的樹
- Boosting 是產生非常多棵的樹但是每一棵的樹都很簡單

目標把多個簡單的樹合再一起才能當最後的預測

Bagging vs. Boosting

一般來說 Boosting 的模型會比 Bagging 來的精準。

- Bagging 透過抽樣的方式生成樹，每棵樹彼此獨立
- Boosting 透過序列的方式生成樹，後面生成的樹會與前一棵樹相關

Part 2

XGBoost (迴歸器)
程式實作

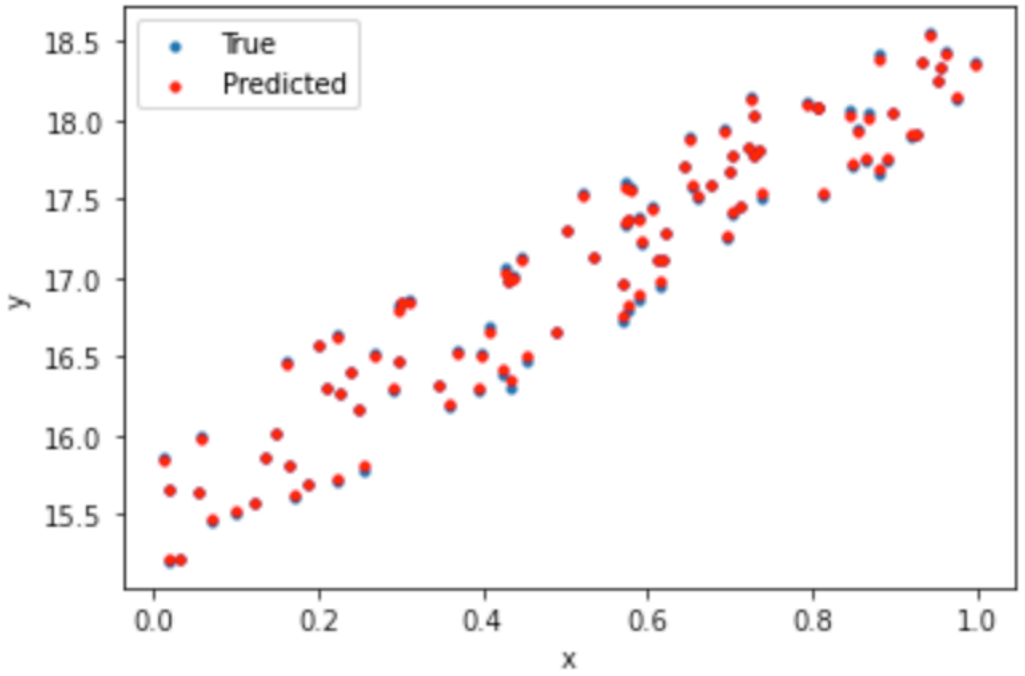
第12屆 iT邦幫忙 鐵人賽

XGBoost Regressor 簡單線性迴歸

```python
import xgboost as xgb

# 建立xgbrModel模型
xgbrModel=xgb.XGBRegressor()
# 使用訓練資料訓練模型
xgbrModel.fit(x,y)
# 使用訓練資料預測
predicted=xgbrModel.predict(x)
```

- 目標函式: $y=3x+15$
- 隨機添加 noise 讓資料分散
- x值域介於 0~1

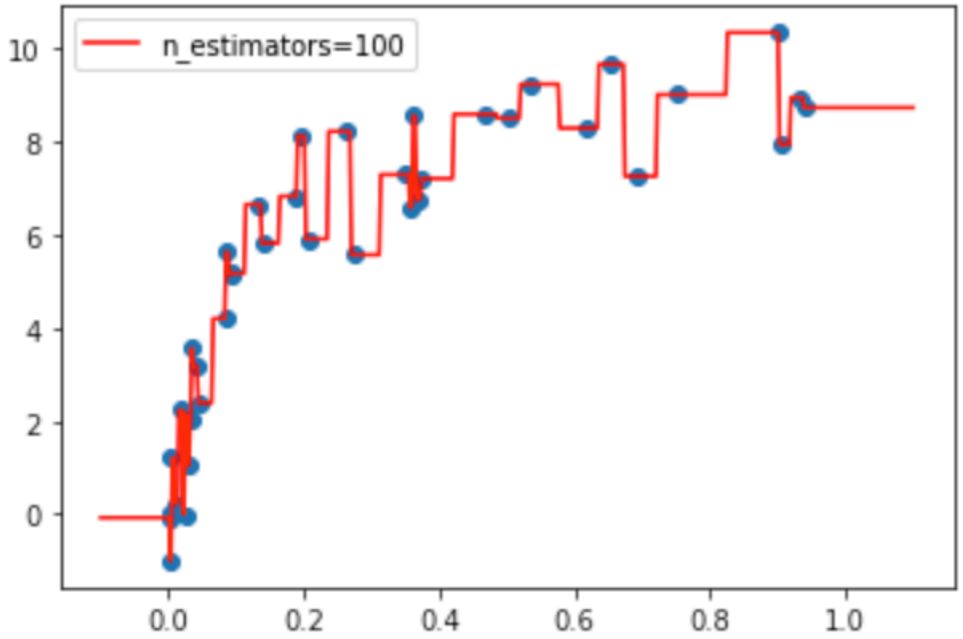


非線性迴歸

動手試試看調整不同的 n_estimators 是否能夠提升預測。

Parameters:

- n_estimators: 總共迭代的次數，即決策樹的個數。預設值為100。
- learning_rate: 學習速率，預設0.3。
- gamma: 懲罰項係數，指定節點分裂所需的最小損失函數下降值。



Thanks

PRESENTED BY 10程式中

第12屆 iT邦幫忙 鐵人賽