

Day 46

機器學習

梯度提升機 - 程式碼撰寫



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(
```

```
    loss="deviance", #Loss 的選擇，若改為 exponential 則會變成
```

```
Adaboosting 演算法，概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
```

```
)
```


解題時間 It's Your Turn

請跳出PDF至官網Sample Code & 作業
開始解題

