前情提要

Ensemble

- 利用多個同質或不同質的Model來**共同**解決一個問題，而各個Model間需具有差異性

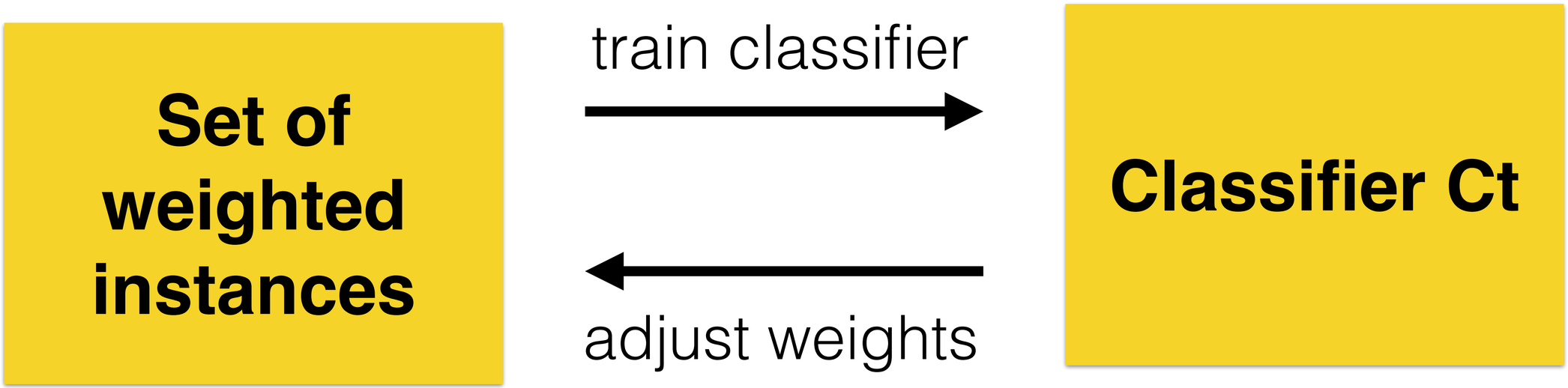
- 關鍵在於**獲得好的Model**及將這些Model以一種**合適的方式“集成”起來**

Bagging

- 同一種Model用**不同的樣本集合**訓練
- 得到多個Model後以投票或平均得到結果，結果將會比單一Model準確率高
- 適合用於較”不穩定”的Model，Bagging能使結果更穩定

Boosting

- 通過數據分佈改變來實現，分類正確樣本降低權重，
  相反的分類錯誤增加權重，亦可用來鑒別異常樣本。



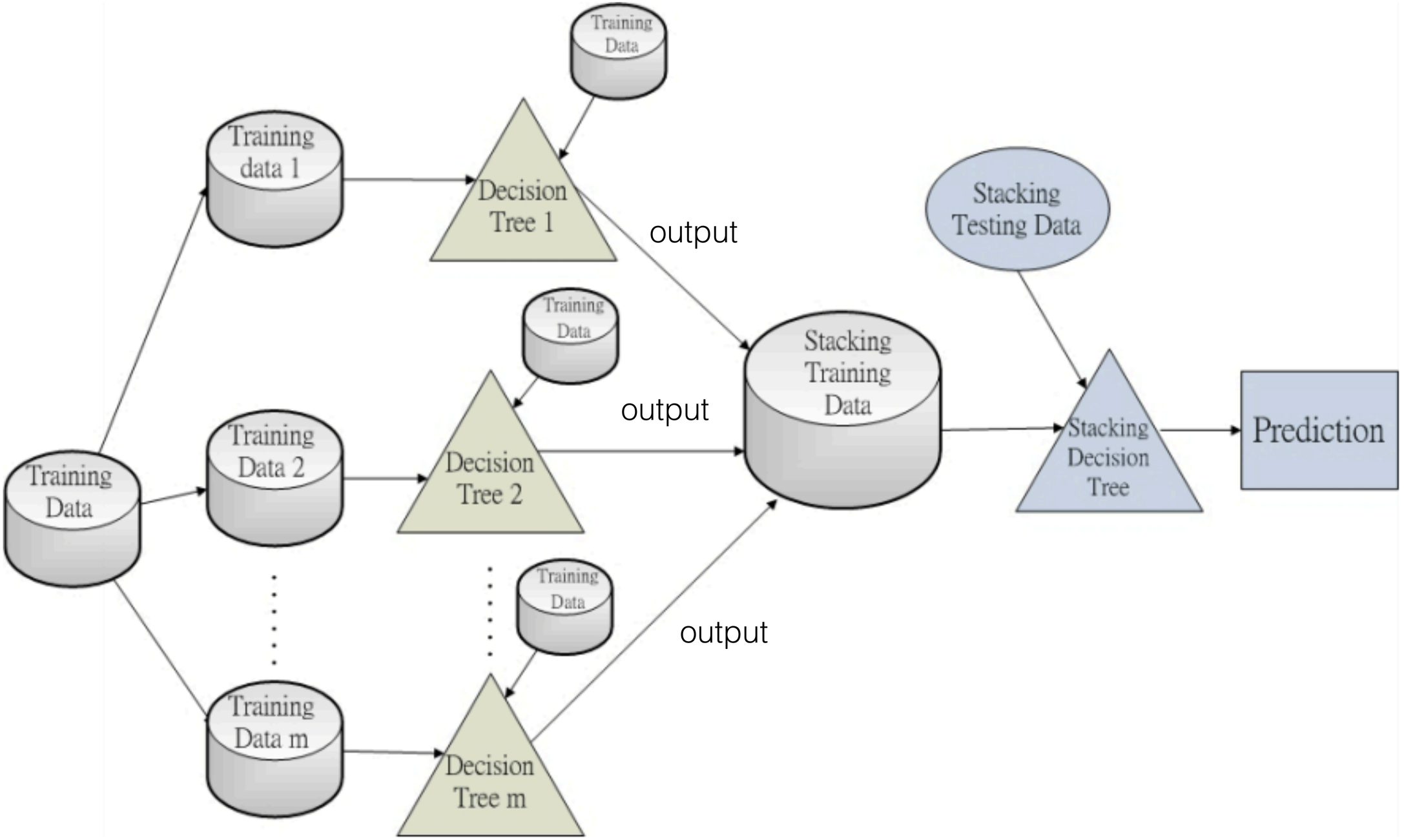
Bagging vs Boosting

改變訓練集合	改變訓練集合
隨機選取，訓練集彼此獨立	不獨立，和前輪學習結果有關
預測可並行產生	預測只能順序產生
投票沒有權重	投票有權重，準確率越高越重

Stacking

Stacking

- 整合不同的學習演算法
- 將不同學習演算法的輸出當成下一層的輸入


Training Data
Training data 1
Decision Tree 1
output
Training Data
Training Data 2
Decision Tree 2
output
Training Data
Training Data m
Decision Tree m
output
Stacking Training Data
Stacking Testing Data
Stacking Decision Tree
Prediction

Training Data

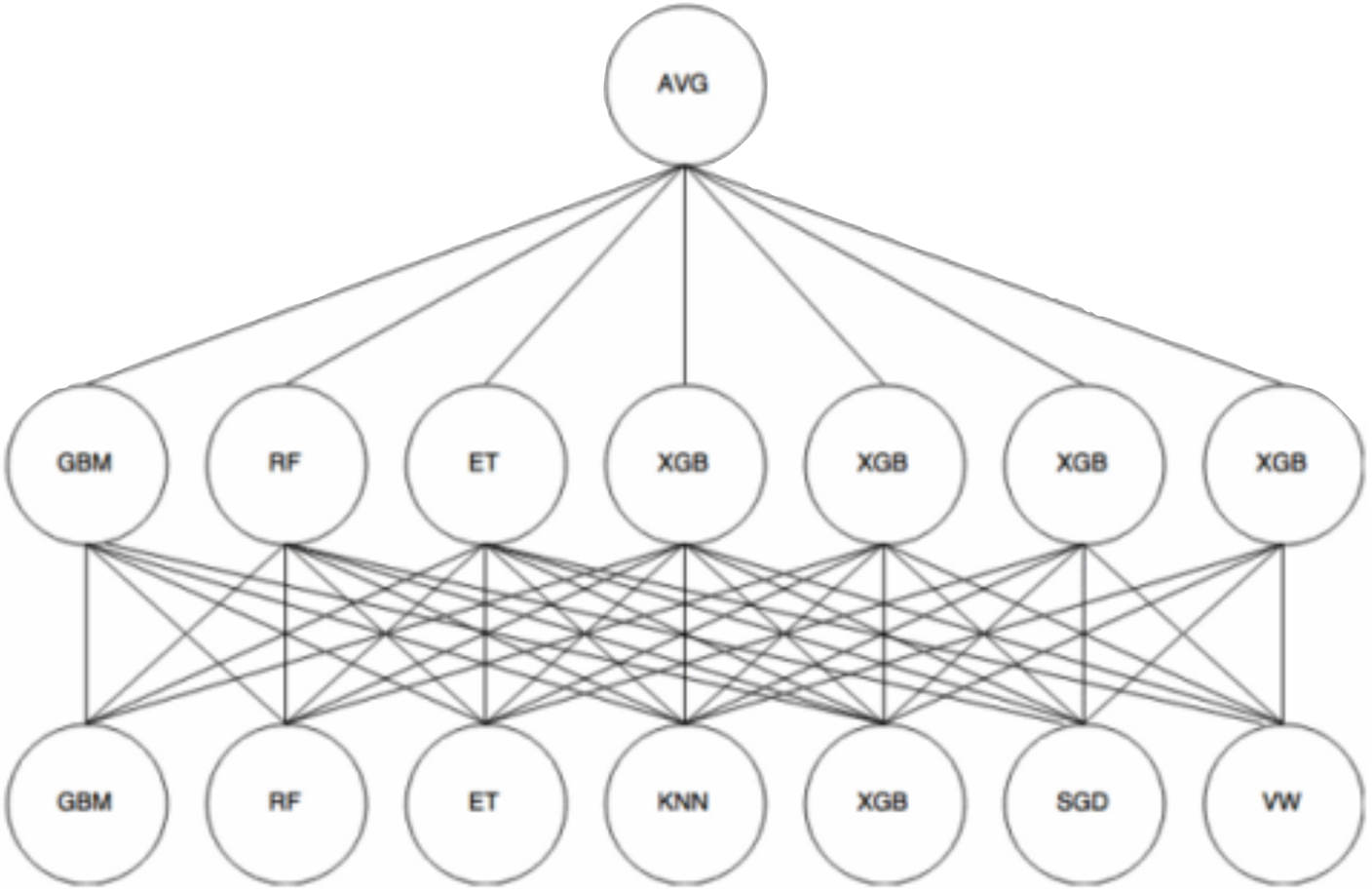
Day	Outlook	Temperature	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No

Stacking Training Data

Day	Decision Tree 1	Decision Tree 2	Decision Tree 3	Decision Tree 4	Decision Tree 5	Play Tennis
D1	Yes	Yes	Yes	No	No	No
D2	Yes	No	Yes	Yes	Yes	No
D3	No	No	No	No	Yes	Yes
D4	No	No	Yes	No	No	Yes
D5	Yes	No	No	No	No	Yes
D6	Yes	No	No	Yes	Yes	No

Stacking

- Stacking 的優勢在於其利用**兩階段的模式**整合不同的預測結果，可能會有翻轉性的結果。
- 第一層學習演算法完全獨立，第二層在第一層的基礎上做預測。
- 高度相關的演算法並不適合。
- 可設計為多層。


AVG
GBM
RF
ET
XGB
XGB
XGB
XGB
GBM
RF
ET
KNN
XGB
SGD
VW

Let’s Design Your Own Stacking


HIVE
APACHE
Spark
Luigi

客戶**Roger**活動事件流

newest

old

Roger

刷了**世界卡**, 消費了**3000元**於**莫爾頓牛排**

2015/05/27 20:48 - 於**忠孝東路XXX 45F**

Roger

兌換了**紅利商品 - 茶葉蛋**

2015/04/29 12:08 - 於**MyRewards**

Roger

開辦了**信用卡戶 - 長榮無限**

2015/02/29 9:34 - 於**信義分行**

ACTOR_TYPE	ACTOR_ID	ACTION_TYPE	ACTION_TIME	OBJECT_TYPE	OBJECT_ID	CHANNEL_TYPE	CHANNEL_ID
customer	$CUSTOMER_ID	$TXN_CODE * 一般消費(40, 41) * 預借現金(30, 31)	$TXN_DATE 被丟掉的 分期, 繳費	merchant: 特店	$MERCHANT_NBR	credit card	$CARD_NBR

ATTRIBUTES

```
{
    "action": {
        "txn_amt": "交易金額",
        "original_currency_code": "原幣別",
        "total_installment_times": "分期總期數",
        "purchase_type_code": "交易分類",
        "consumption_category_desc": "MCC code_消費類別"
    },
    "object": {
        "merchant_category_code": "MCC code",
        "merchant_name": "商店名"
    },
    channel: {
        "card_type": "信用卡種"($Kind1),
        "card_level": "信用卡等"($Kind2)
    }
}
```

ACTOR_TYPE	ACTOR_ID	ACTION_TYPE	ACTION_TIME	OBJECT_TYPE	OBJECT_ID	CHANNEL_TYPE	CHANNEL_ID
customer	$CUSTOMER_ID	$TXN_CODE * 一般消費(40, 41) * 預借現金(30, 31)	$				

ATTRIBUTES

```
{
    "action": {
        "txn_amt": "交易金額",
        "original_currency_code": "原幣別",
        "total_installment_times": "分期總期數",
        "purchase_type_code": "交易分類",
        "consumption_category_desc": "MCC code_消費類別"
    },
    "object": {
        "merchant_category_code": "MCC code",
        "merchant_name": "商店名"
    },
    channel: {
        "card_type": "信用卡種"($Kind1),
        "card_level": "信用卡等"($Kind2)
    }
}
```

+

+

- Spark Save HDFS File不能以Partition存放問題
- 環境變數設定：須確認SPARK路徑及Hadoop路徑
- HDFS需使用完整路徑，包含nameservice
- 存成HDFS檔案需確認Partition是否存在