

心理與神經資訊學

(Psychoinformatics & Neuroinformatics)

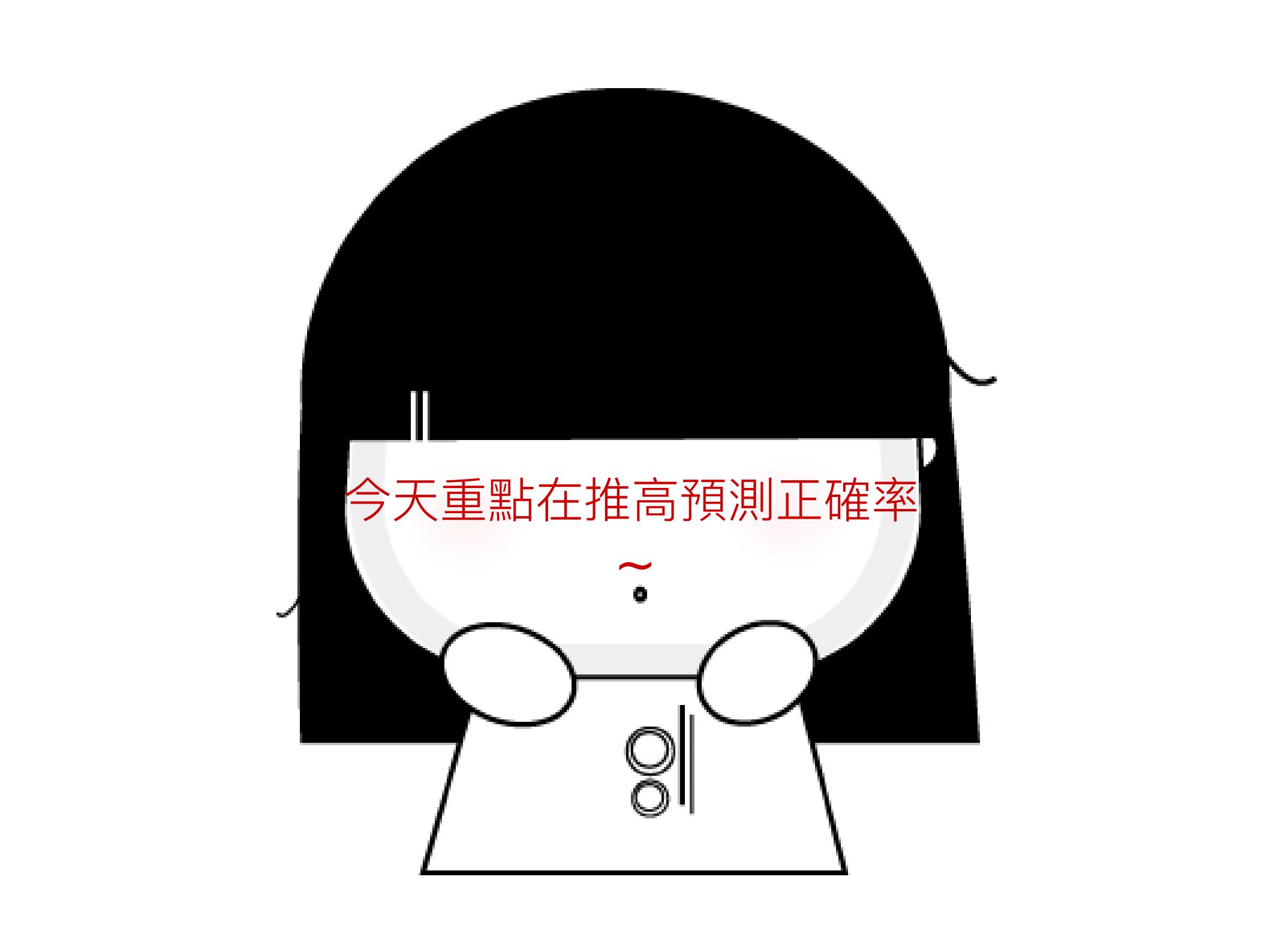
課號：Psy5261

識別碼：227U9340

教室：博雅 101

時間：四 234





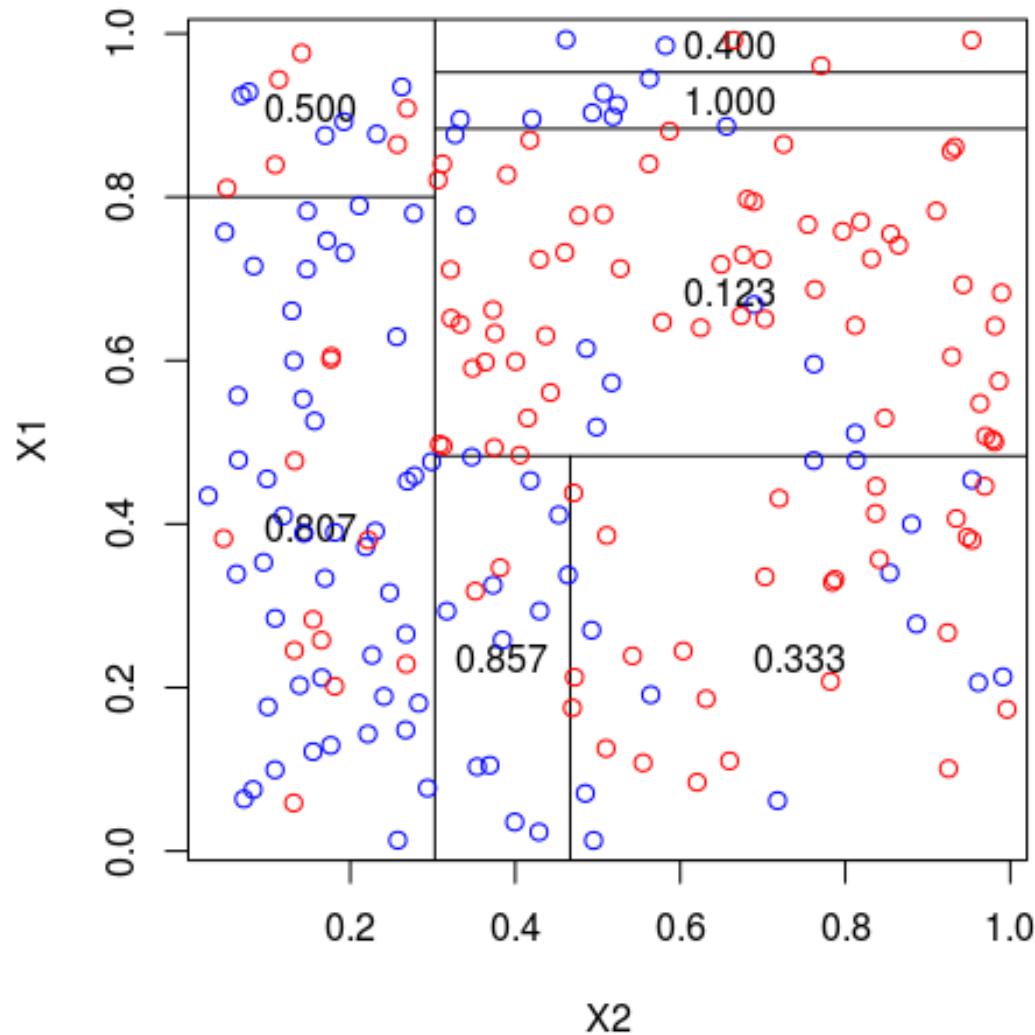
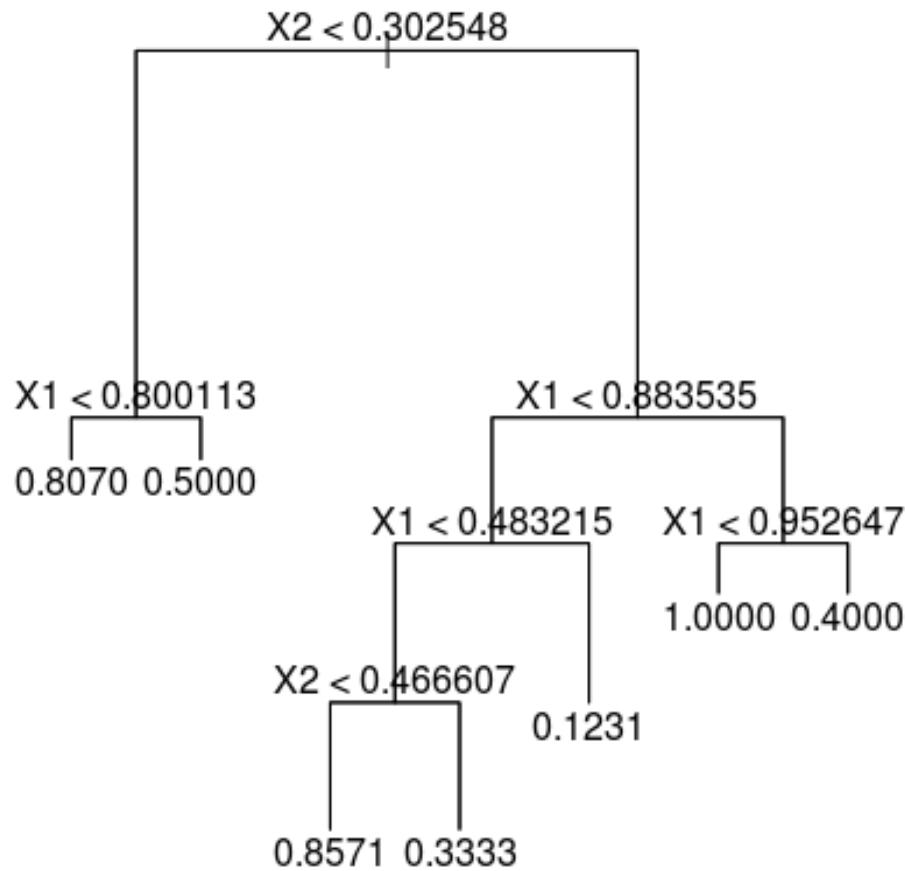
今天重點在推高預測正確率

監督式學習

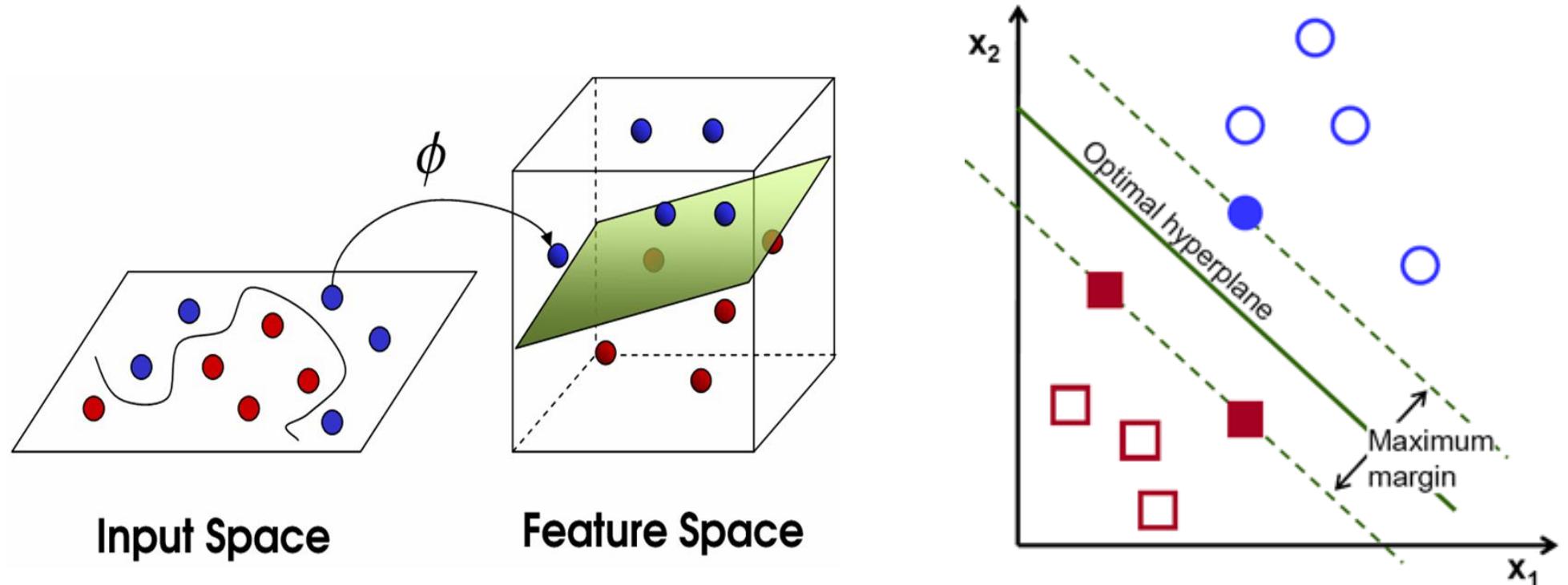
(Supervised Learning)

From Classification to Regression

分類和迴歸問題是一體的兩面



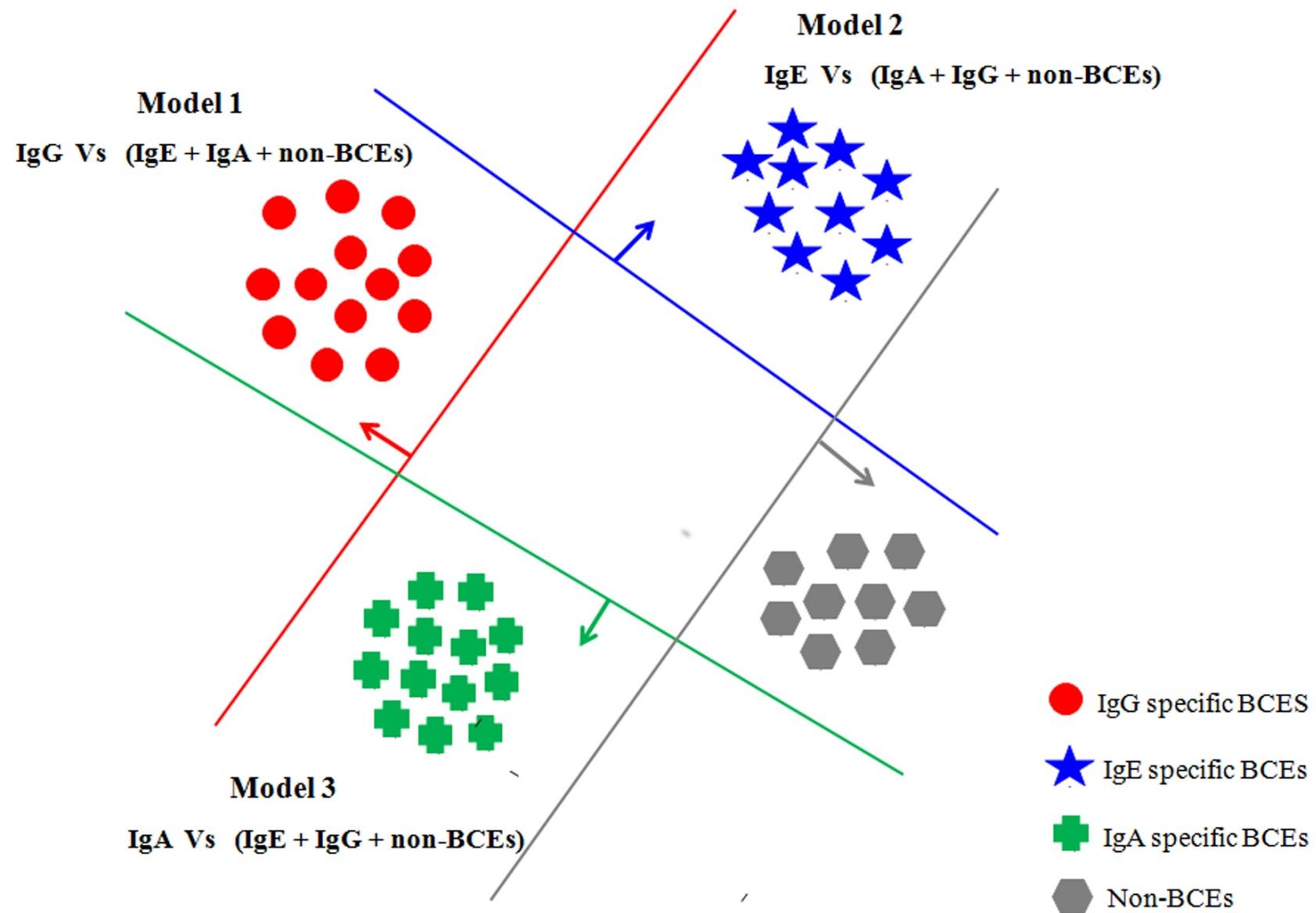
Support Vector Machine



```
from sklearn import *
clf=svm.SVC()
clf.fit(X,Y) #training
print(np.mean(clf.predict(X)==Y)) #testing
```

Multiclass Classification

One vs. One 比 One vs. Rest 均衡但費時



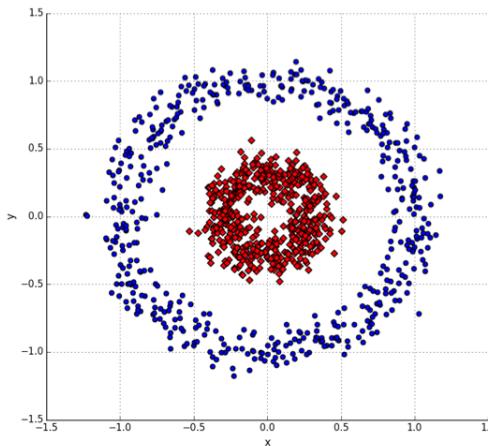
Multilabel Classification

又稱為 multi-output classification

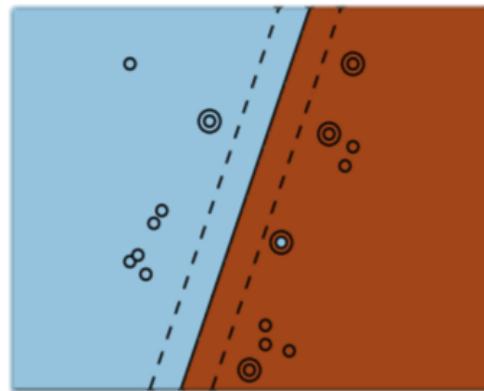
Images			
Labels	tree water black picture drawing sea art blue boat green city	man woman people hair girl picture smile group photo kid family	sky tree water white house window wood sea ocean cloud blue door
MLR-GL	<i>man white black woman people blue green red tree girl sky water hair picture old brown grass yellow face mountain</i>	man woman black white people blue green red girl tree hair sky water picture old brown face yellow grass smile	<i>man black white woman blue people green red girl tree sky hair water picture old brown yellow face grass window</i>
LIBSVM	<i>book smile gray sun flag computer brick man yellow street machine sea leaf road ocean couple forest fly purple toy</i>	man hair black movie face food fire boy smile lady metal statue dance couple red table toy arm bike gold	<i>building sky fence floor church shirt legs wall money glass ship room couple word city bald door guy orange chart</i>
LIBSVM+Platt	<i>book man smile white blue sky black woman red green people tree water computer girl face old hair yellow leaf</i>	<i>movie food man hair white smile woman blue face black people green red girl fire tree sky boy table eye</i>	<i>building sky man red floor white black woman blue church fence people green hair tree face shirt room grass chart</i>
MLR-L1	tree green hair movie white black people grass statue leaf orange old bike red flower mountain picture dance eye dirt	hair tree black movie green man eye woman white hand face girl people smile dance red hat orange statue brown	tree hair movie black white green square people eye blue dance hand hat orange logo wall red statue man bike

HyperParameters

邊界的位置是 parameters ; 模型結構是 hyperparams

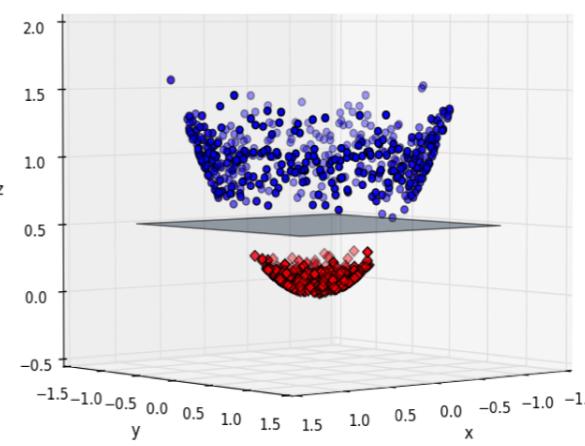


Linear Kernel

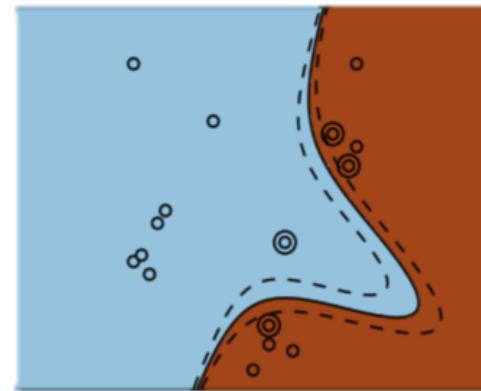


C hyperparameter

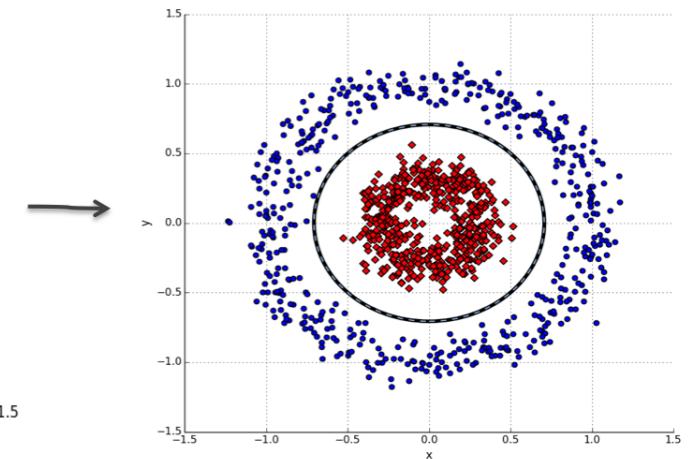
φ



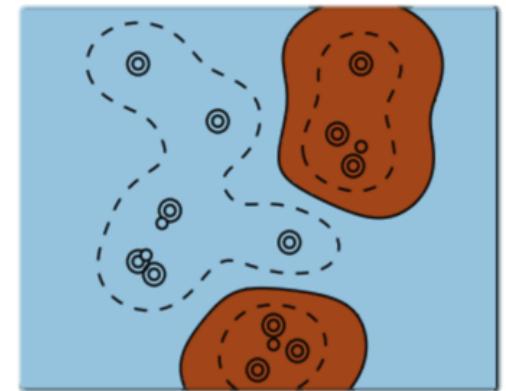
Polynomial Kernel



C plus gamma, degree and coefficient hyperparameters



RBF Kernel

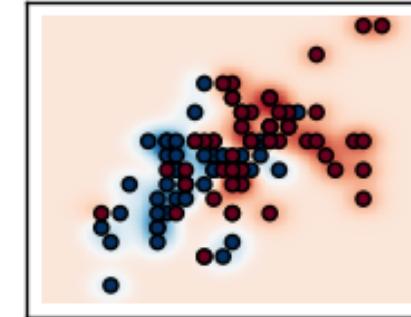
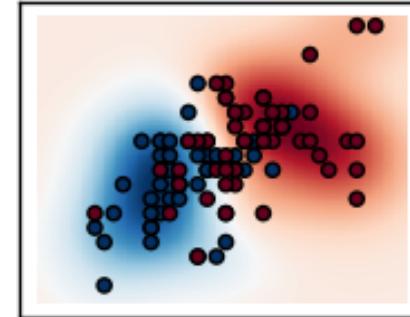
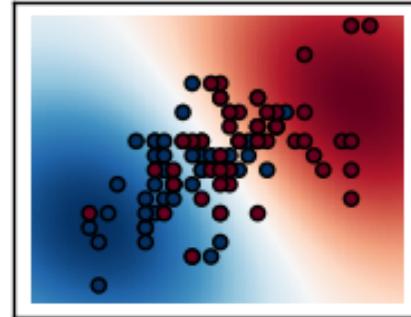


C plus gamma hyperparameter

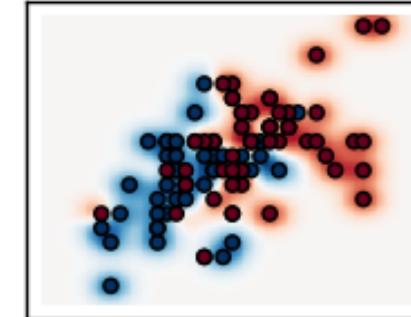
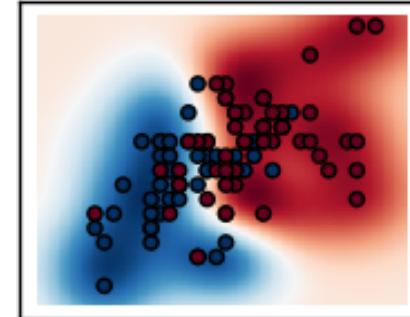
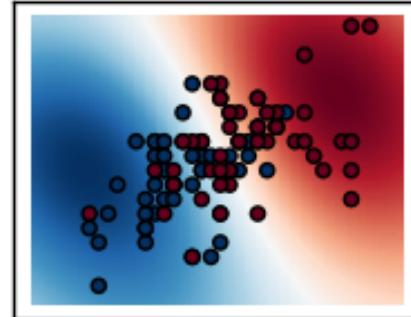
Performance Tuning: Grid Search

就是有系統化地改變 hyperparams 最佳化模型表現

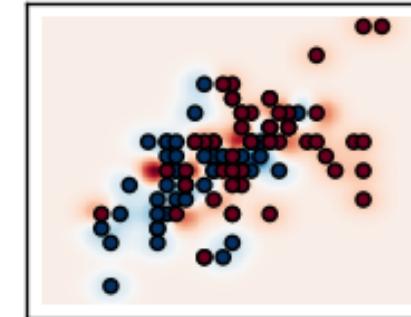
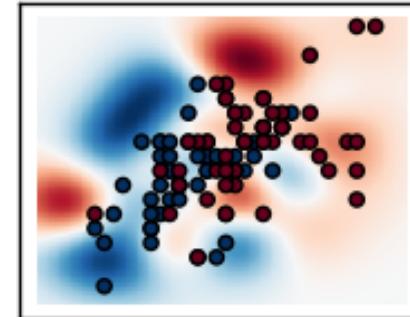
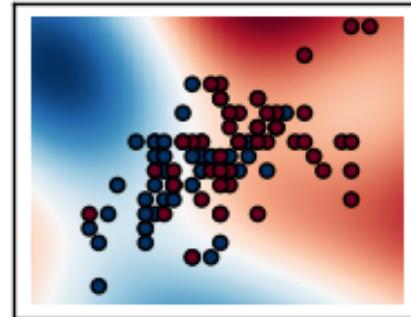
gamma=10⁻¹, C=10⁻² gamma=10⁰, C=10⁻² gamma=10¹, C=10⁻²



gamma=10⁻¹, C=10⁰ gamma=10⁰, C=10⁰ gamma=10¹, C=10⁰



gamma=10⁻¹, C=10² gamma=10⁰, C=10² gamma=10¹, C=10²

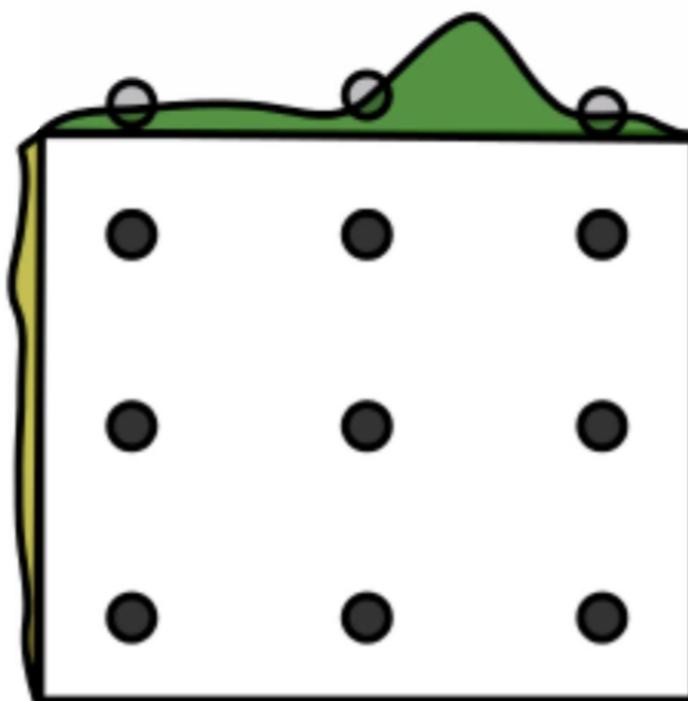


Performance Tuning: HyperOpt

Grid Search 系統化但搜尋效率輸 Random Search

Grid Layout

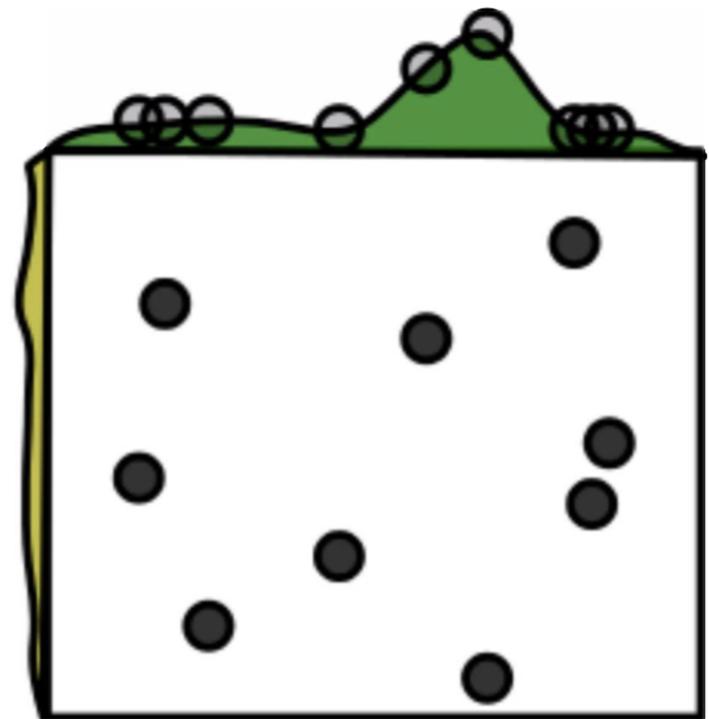
Unimportant parameter



Important parameter

Random Layout

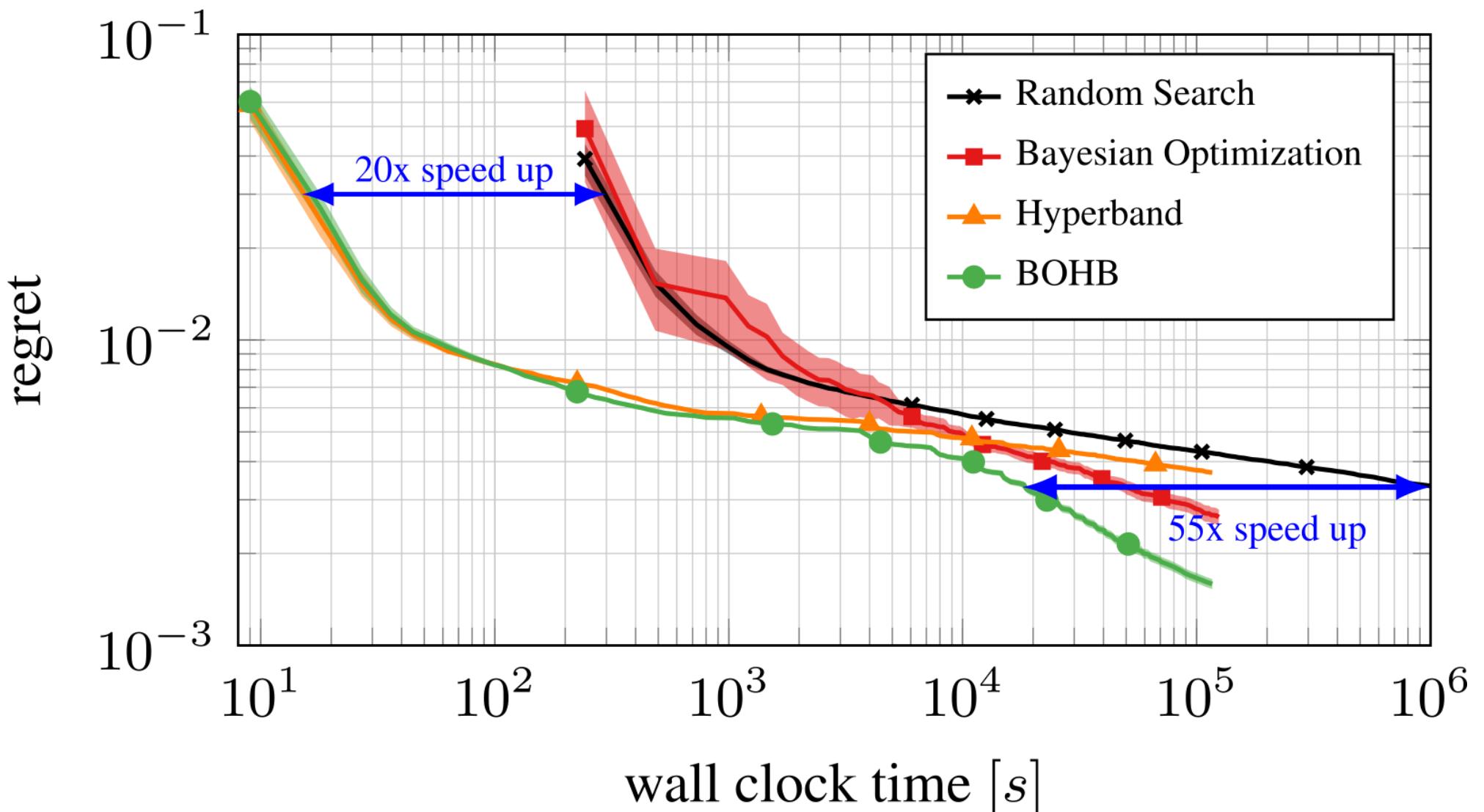
Unimportant parameter



Important parameter

Performance Tuning: HyperBand

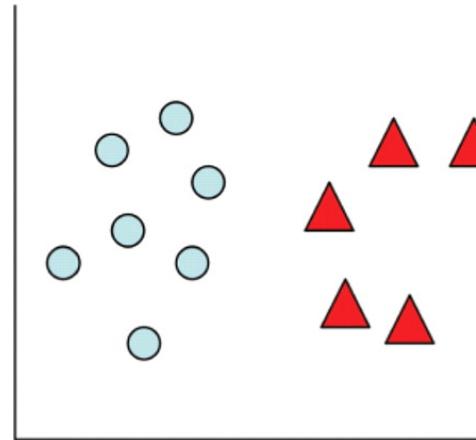
State-of-the-art: HyperBand family



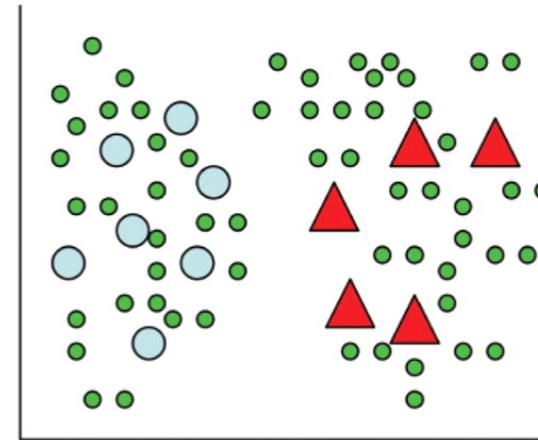
**半監督式學習
(Semi-supervised Learning)**

Semi-supervised Learning

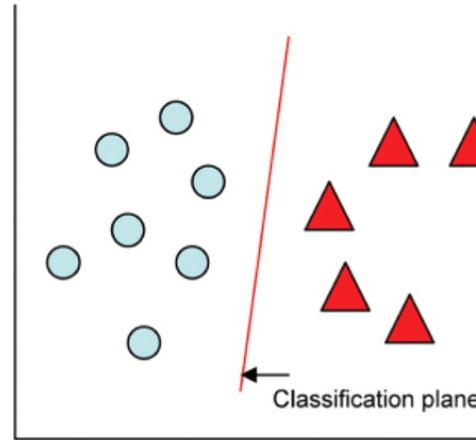
可利用 sklearn.semi_supervised



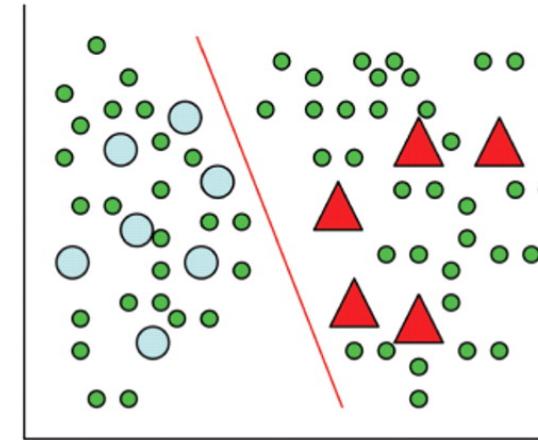
Labeled Data
(a)



Labeled and Unlabeled Data
(b)



Supervised Learning
(c)



Semi-Supervised Learning
(d)

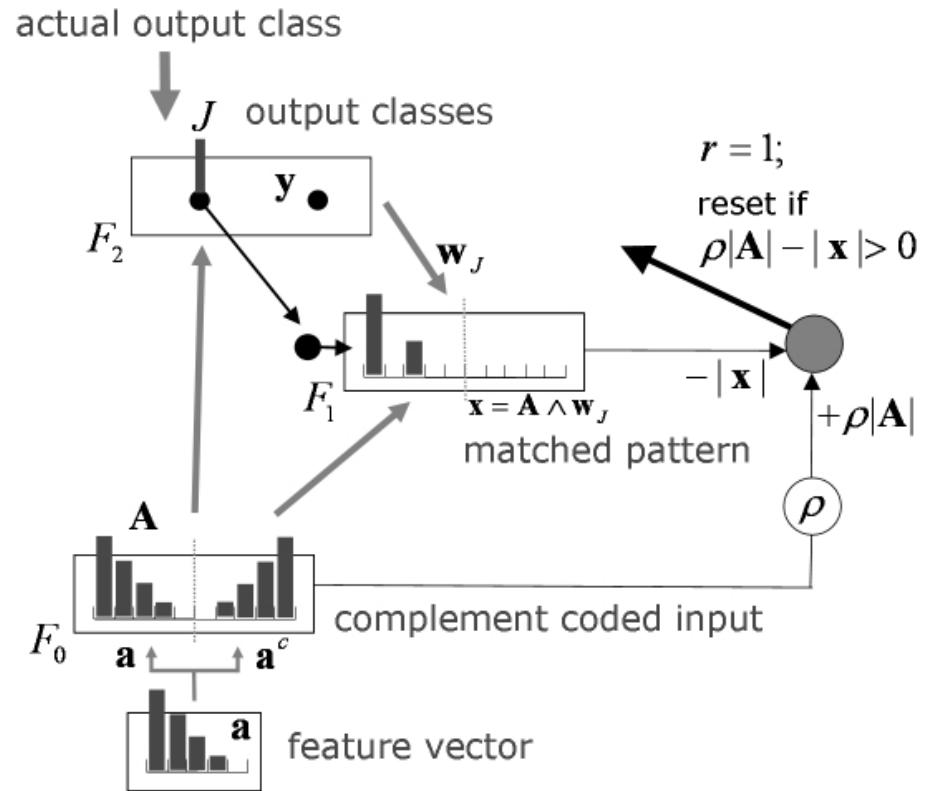
混合式學習

(Hybrid/Mixed Learning)

Hybrid Learning (1/3)

不同學習法可以混搭

如 ARTMAP 中先做
unsupervised learning 再
做 supervised learning:



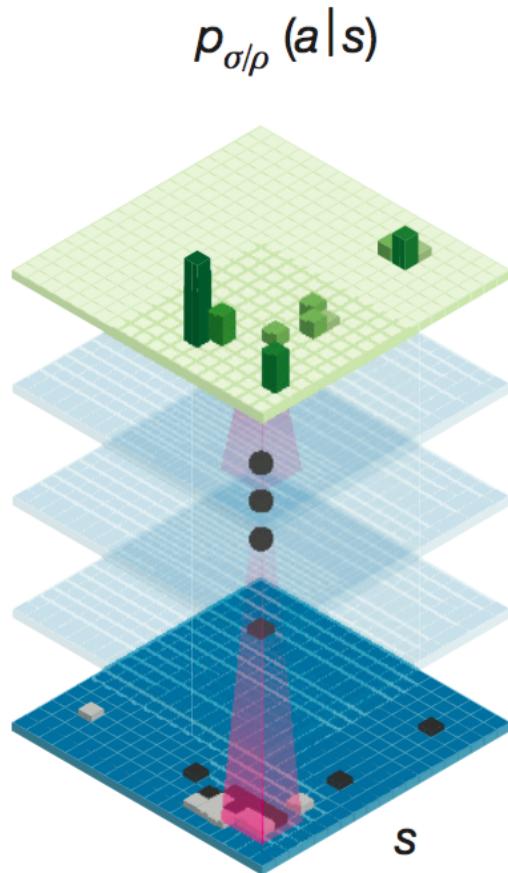
也可利用 ML 中的基因演算法去最佳化其他 ML 模型參數：

- **步驟 1:** 自行定義 fitness function $y=f(\mathbf{X})$
(\mathbf{X} 為參數向量； f 執行模型評估； y 為正確率 / 錯誤率)
- **步驟 2:** 將 fitness function f 傳給基因演算法來最佳化

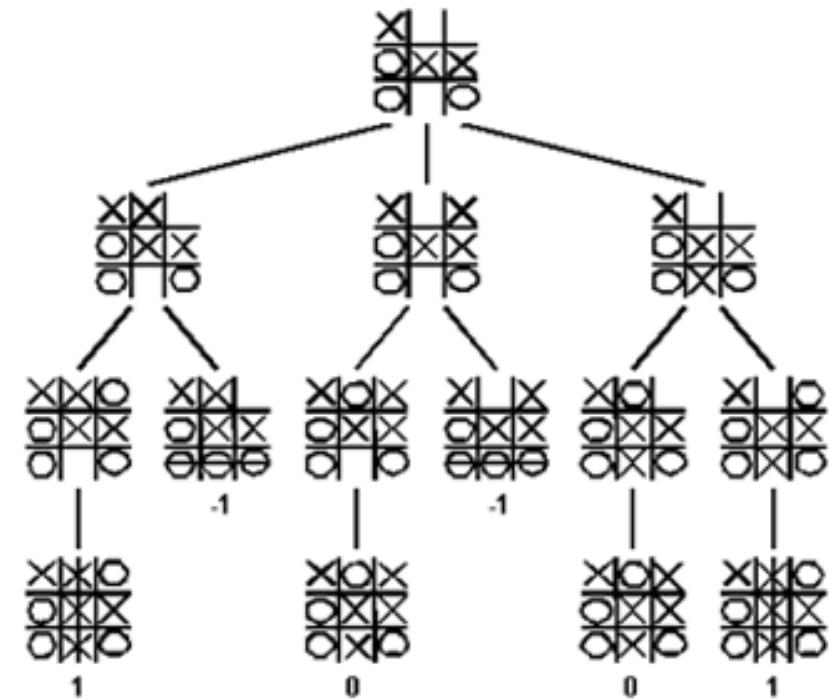
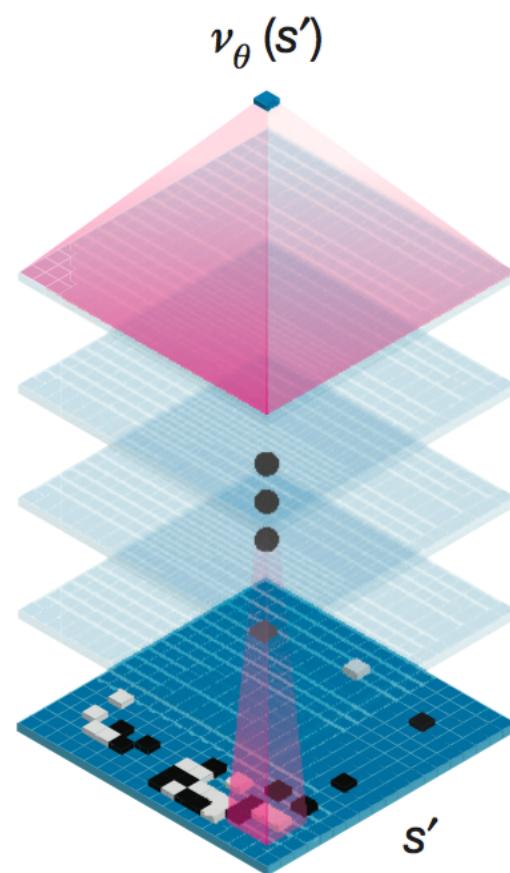
Hybrid Learning (2/3)

左是監督式學習；右是增強式學習

Policy network



Value network



Hybrid Learning (3/3)

NEURAL ARCHITECTURE SEARCH WITH REINFORCEMENT LEARNING

Barret Zoph*, Quoc V. Le

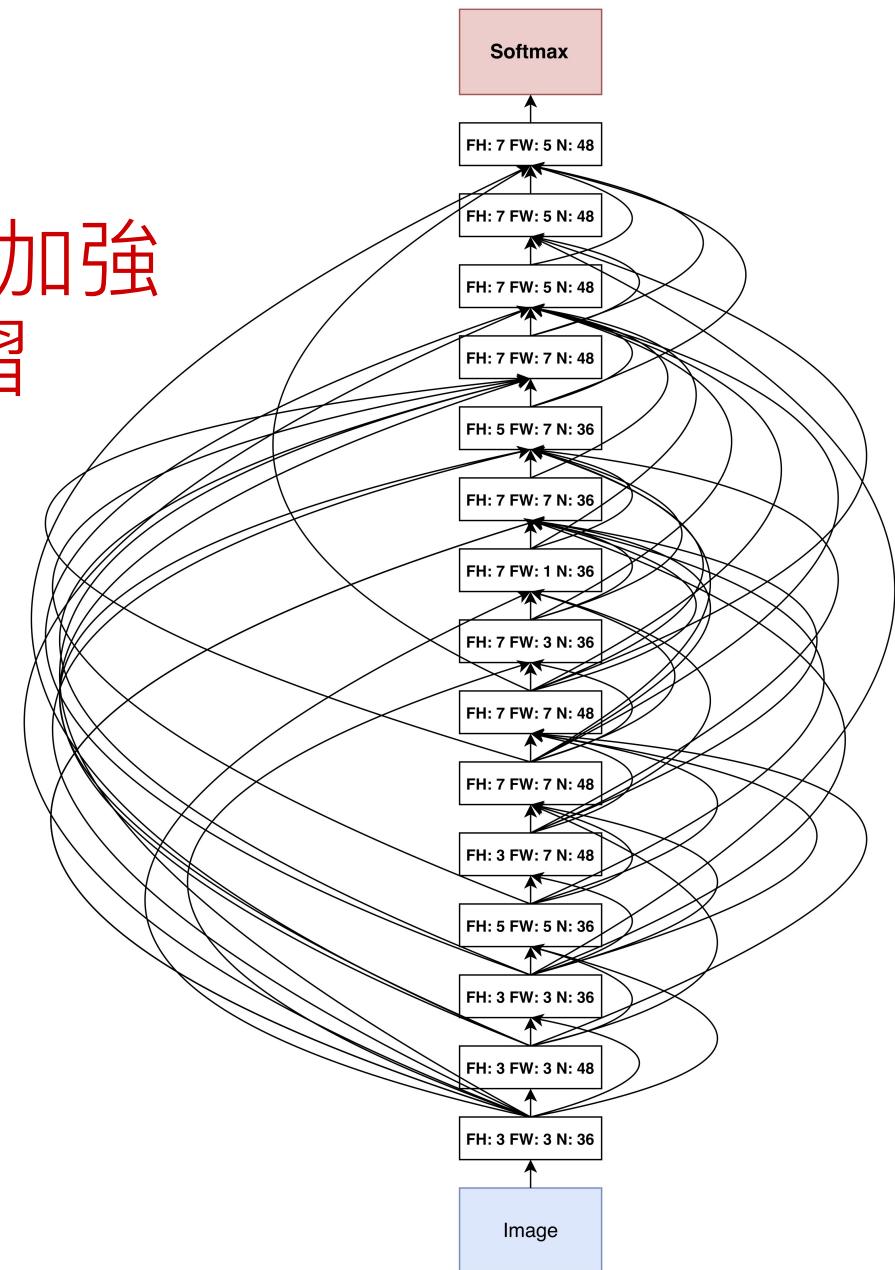
Google Brain

{barrettzoph, qvl}@google.com

用增強式學習加強
監督式學習

ABSTRACT

Neural networks are powerful and flexible models that work well across learning tasks in image, speech and natural language under their success, neural networks are still hard to design. In this paper, we propose a recurrent neural network to generate the model descriptions of neural networks. We train this RNN with reinforcement learning to maximize the expected accuracy of the generated architectures on a validation set. On the CIFAR-10 dataset, our model can start from scratch, can design a novel network architecture that achieves better accuracy than the previous state-of-the-art model. Our model also achieves a test error rate of 3.65, which is 0.09 percent better and faster than the previous state-of-the-art model that used a similar architecture. On the Penn Treebank dataset, our model can compose a novel recurrent neural network that performs the widely-used LSTM cell, and other state-of-the-art baselines. Our model achieves a test set perplexity of 62.4 on the Penn Treebank, which is 1.2 percent better than the previous state-of-the-art model. The cell can also be applied to the character language modeling task on PTB and achieves a test perplexity of 1.214.



Ensemble Learning

就是集結一群 machine learners 來做決策

Common Types of Ensemble Methods

Bagging

- Reduces variance and increases accuracy
- Robust against outliers or noisy data
- Often used with Decision Trees (i.e. Random Forest)

Boosting

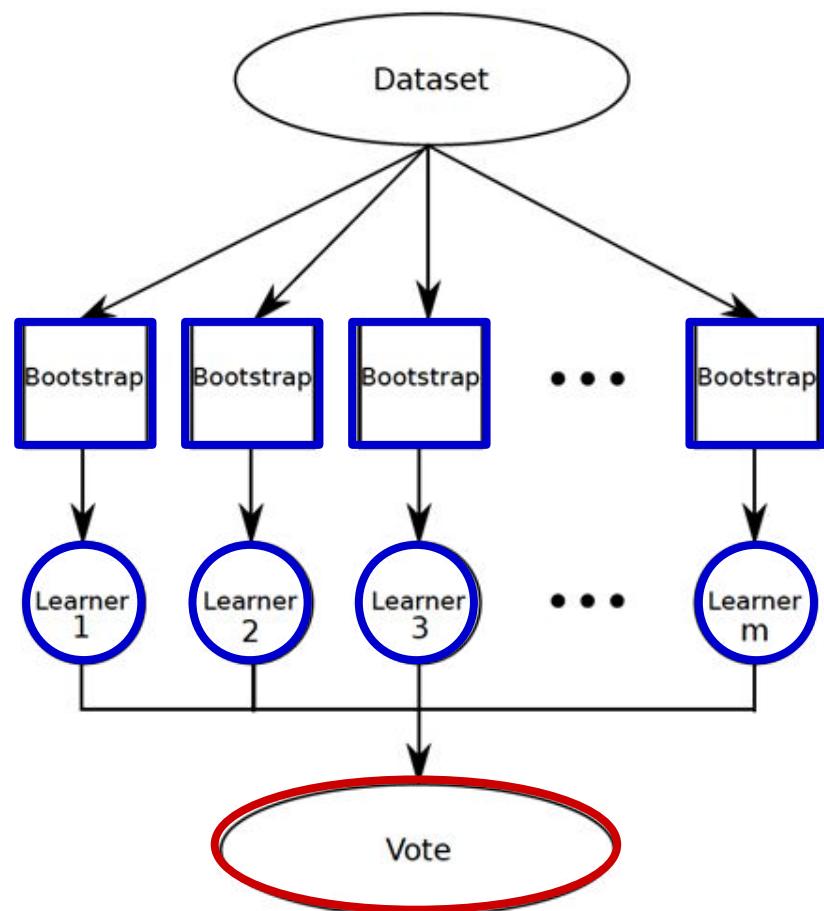
- Also reduces variance and increases accuracy
- Not robust against outliers or noisy data
- Flexible – can be used with any loss function

Stacking

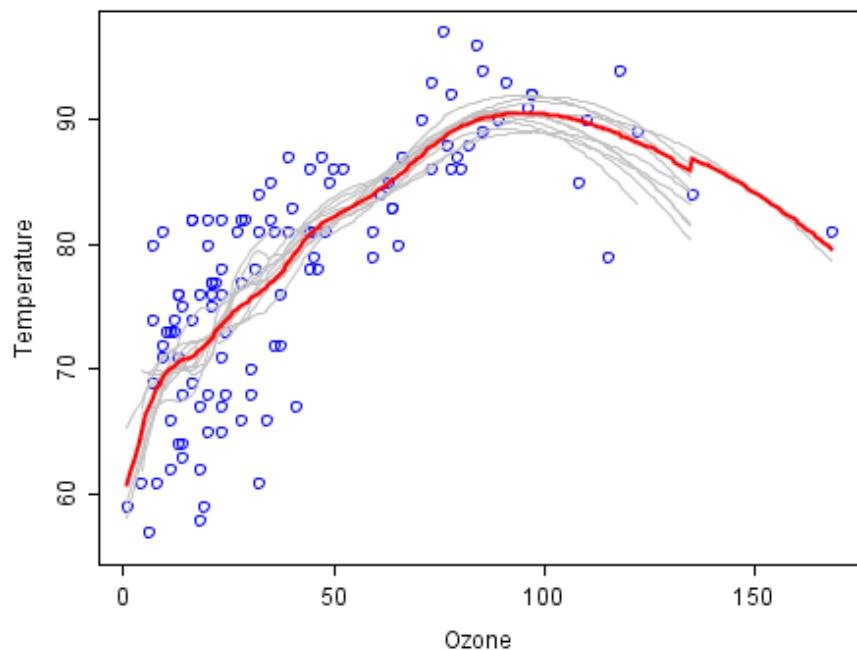
- Used to ensemble a diverse group of strong learners
- Involves training a second-level machine learning algorithm called a “metalearner” to learn the optimal combination of the base learners

Bagging

(同類的)machine learners 學不同的抽樣資料



人生的經驗不同，
做出來的判斷也不同。



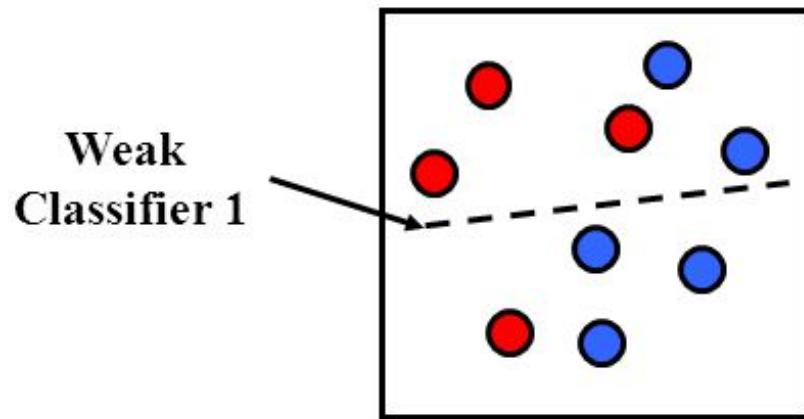
Boosting (1/3)

看起來有兩大類；但 AB 其實是 GB 的特例

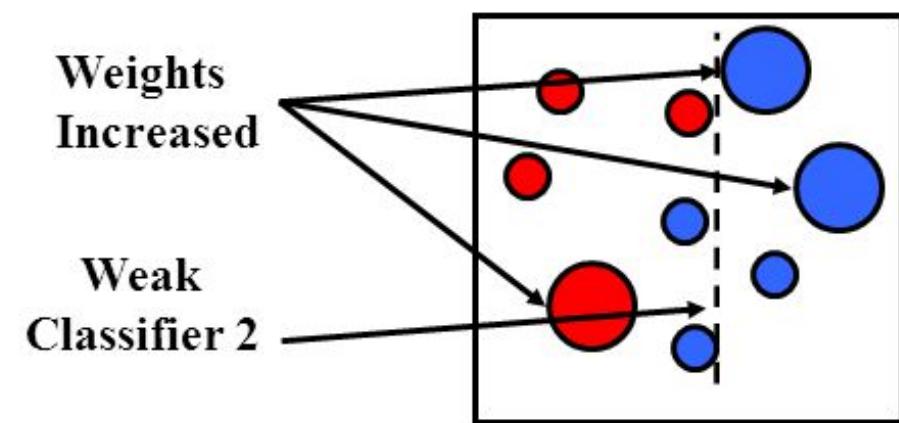
AdaBoost	GradientBoost
Both AdaBoost and Gradient Boost use a base weak learner and they try to boost the performance of a weak learner by iteratively shifting the focus towards problematic observations that were difficult to predict. At the end, a strong learner is formed by addition (or weighted addition) of the weak learners.	
In AdaBoost, shift is done by up-weighting observations that were misclassified before.	Gradient boost identifies difficult observations by large residuals computed in the previous iterations.
In AdaBoost "shortcomings" are identified by high-weight data points.	In Gradientboost "shortcomings" are identified by gradients.
Exponential loss of AdaBoost gives more weights for those samples fitted worse.	Gradient boost further dissect error components to bring in more explanation.
AdaBoost is considered as a special case of Gradient boost in terms of loss function, in which exponential losses.	Concepts of gradients are more general in nature.

Boosting (2/3): AdaBoost

Re-weighting mis-classified training data;
投票時不同 classifiers 的 weights 也不同

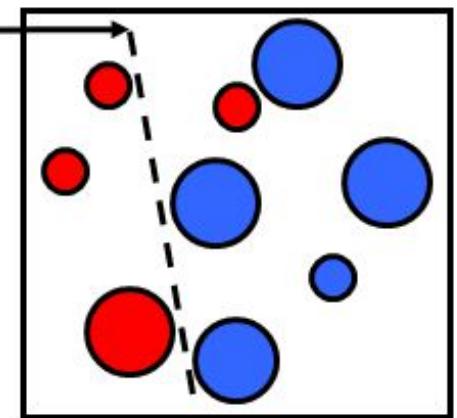


Classifier 1 學不好的繼續交給
Classifier 2 來學 / 處理：



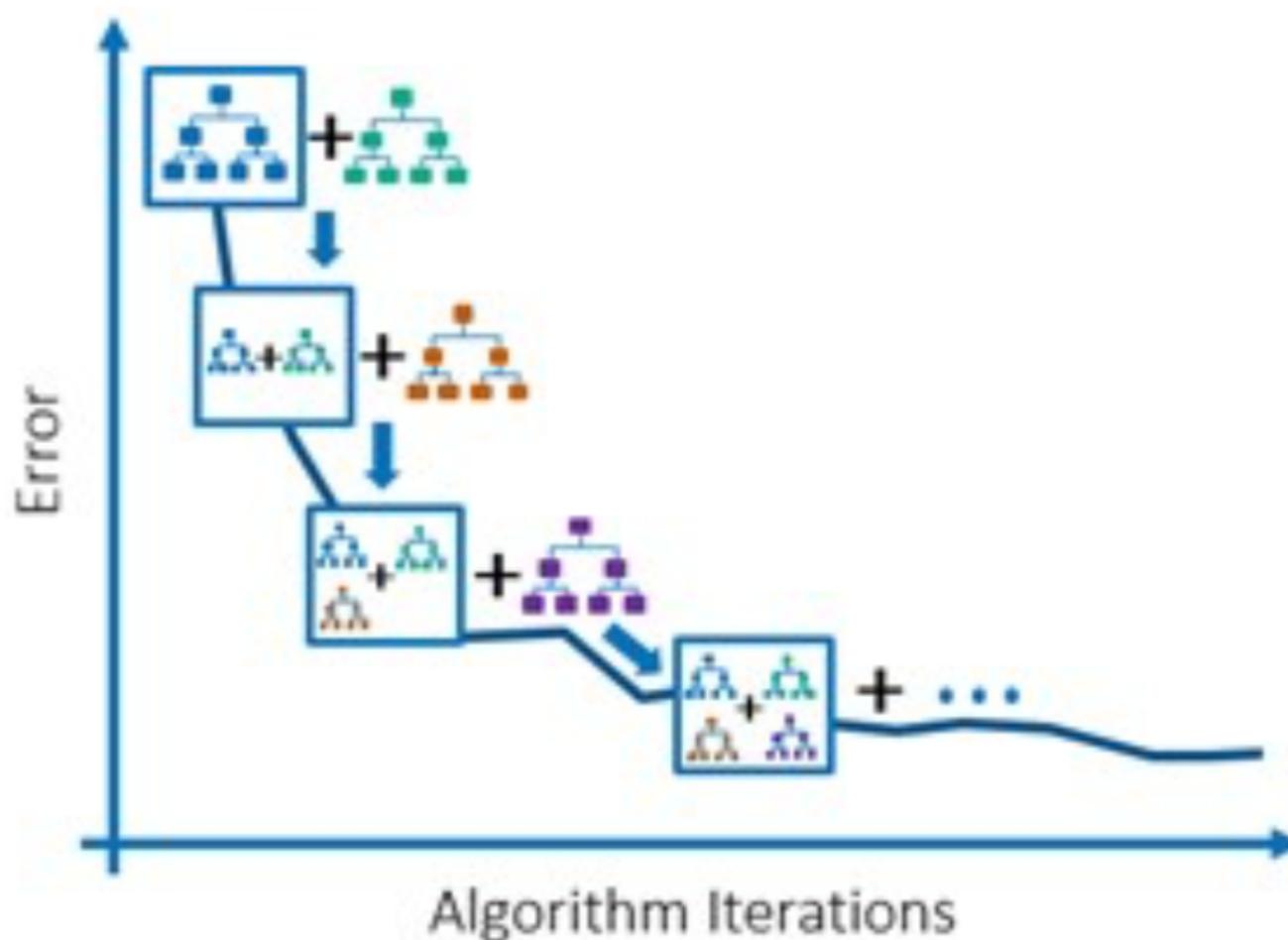
Weak
classifier 3

Final classifier is
linear combination of
weak classifiers



Boosting (3/3): GradientBoost

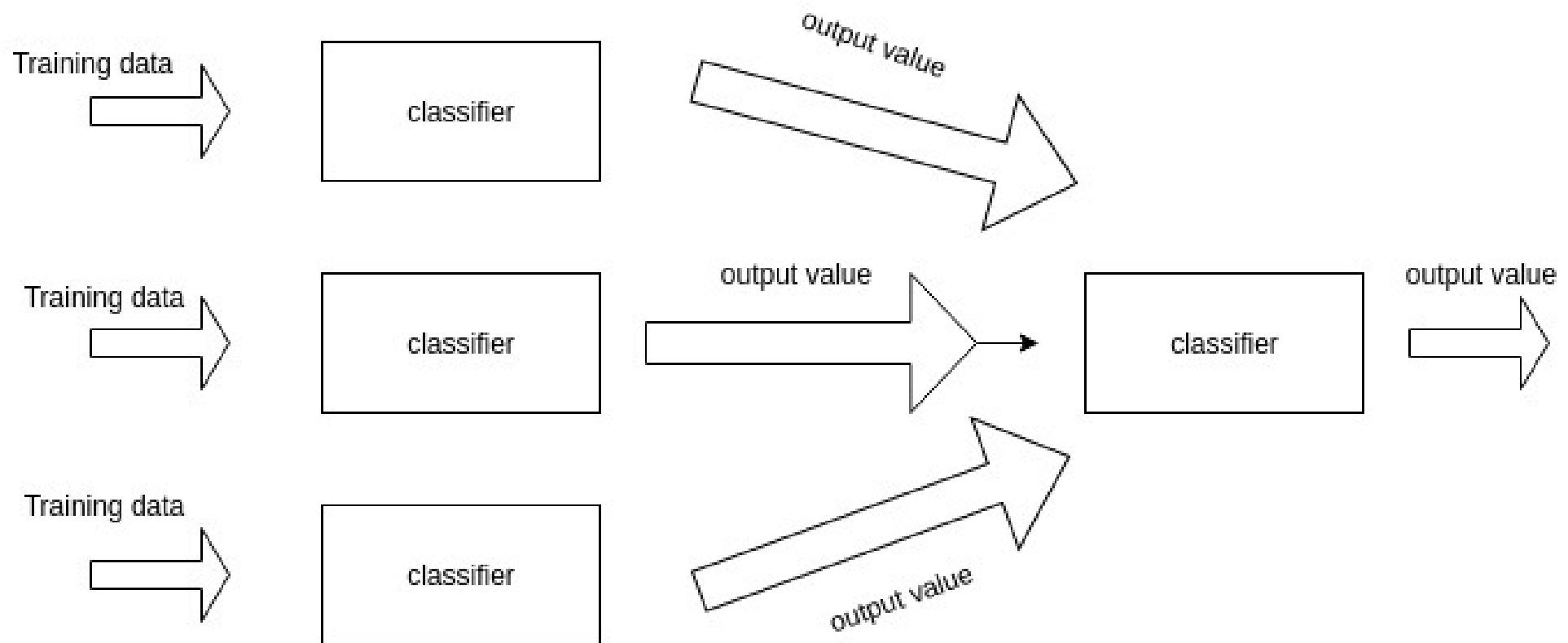
加入後續的 model 是為了要減少整體錯誤率
(c.f. 泰勒展開式)



流行的XGBoost是高效能的 GradientBoost 實作

Stacking

可看成 hierarchical supervised learning



雖 scikit-learn 中沒有直接支援但很容易自己實作

模型評估與選擇 (Model Selection)

Beyond Accuracy

除了正確率外還有很多模型評斷指標

		True condition		Prevalence $= \frac{\sum \text{Condition positive}}{\sum \text{Total population}}$	False discovery rate (FDR) $= \frac{\sum \text{False positive}}{\sum \text{Test outcome positive}}$
Total population		Condition positive	Condition negative		
Predicted condition	Predicted condition positive	True positive	False positive (Type I error)	Positive predictive value (PPV), Precision $= \frac{\sum \text{True positive}}{\sum \text{Test outcome positive}}$	Negative predictive value (NPV) $= \frac{\sum \text{True negative}}{\sum \text{Test outcome negative}}$
	Predicted condition negative	False negative (Type II error)	True negative	False omission rate (FOR) $= \frac{\sum \text{False negative}}{\sum \text{Test outcome negative}}$	
Accuracy (ACC) = $\frac{\sum \text{True positive} + \sum \text{True negative}}{\sum \text{Total population}}$	True positive rate (TPR), Sensitivity, Recall $= \frac{\sum \text{True positive}}{\sum \text{Condition positive}}$	False positive rate (FPR), Fall-out $= \frac{\sum \text{False positive}}{\sum \text{Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR}^+}{\text{LR}^-}$	
	False negative rate (FNR), Miss rate $= \frac{\sum \text{False negative}}{\sum \text{Condition positive}}$	True negative rate (TNR), Specificity (SPC) $= \frac{\sum \text{True negative}}{\sum \text{Condition negative}}$	Negative likelihood ratio (LR-) = $\frac{\text{FNR}}{\text{TNR}}$		

Precision = 提取出的正確信息數 / 提取出的信息數

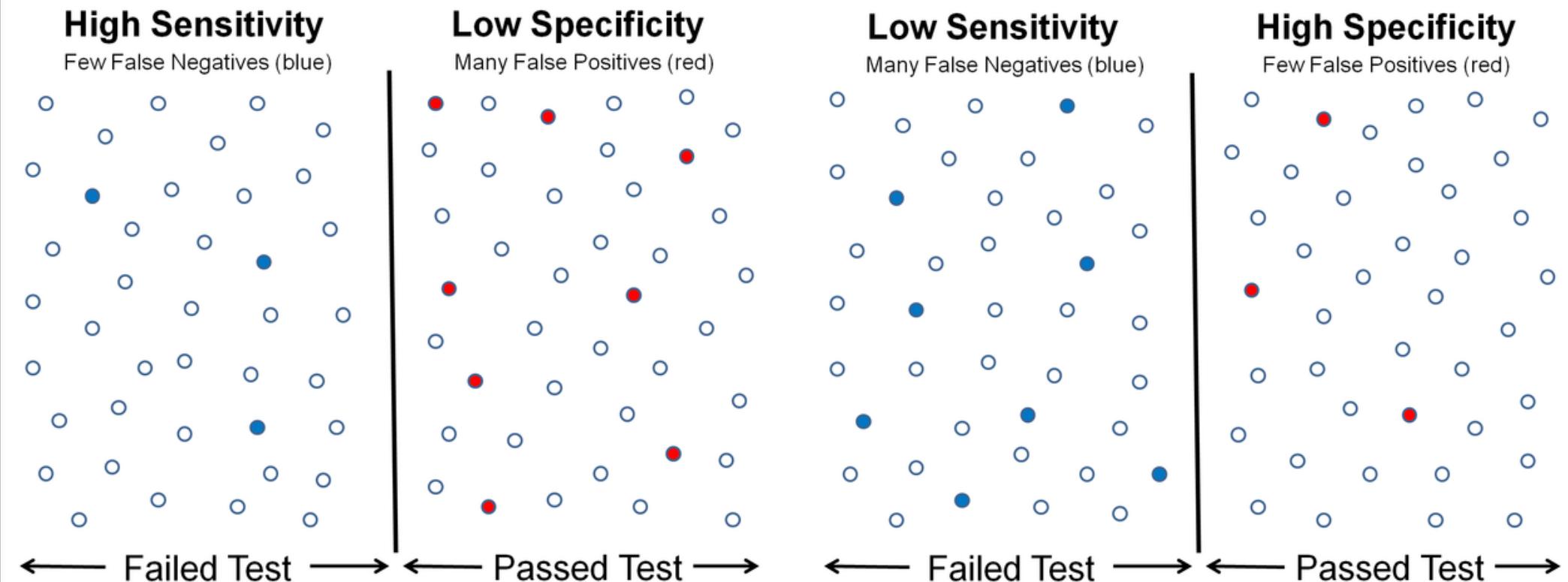
Recall = 提取出的正確信息數 / 樣本的正確信息數

F1 = 兩者的調和平均 = $2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall})$

G = 兩者的幾何平均 = $\sqrt{\text{Precision} * \text{Recall}}$

Sensitivity vs. Specificity

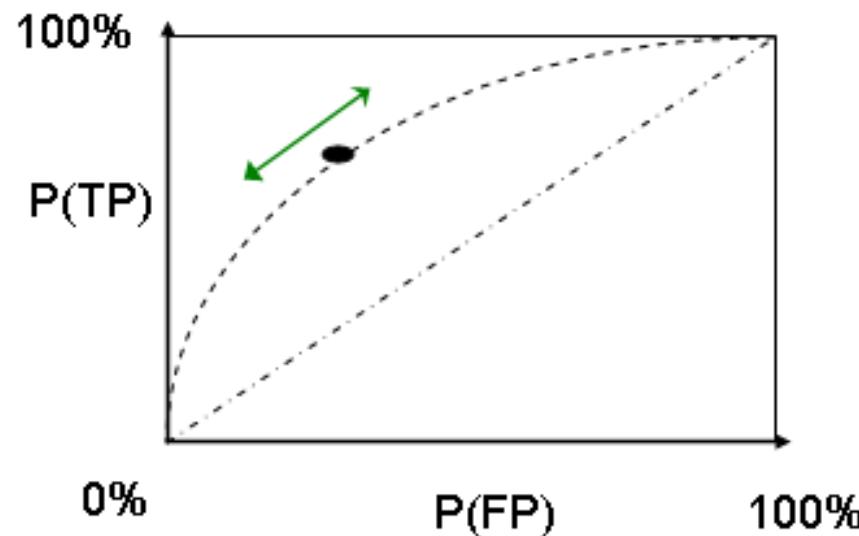
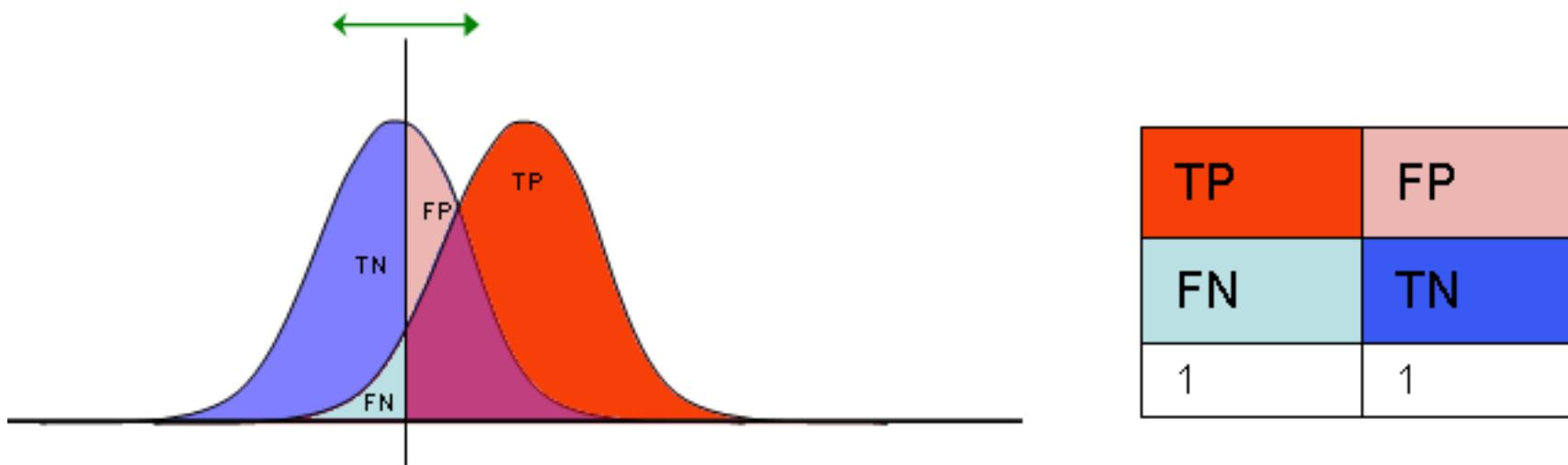
Sensitivity = 預測有 / 真的有
Specificity = 預測沒有 / 真的沒有



Sensitivity 和 Specificity 是一個 trade-off

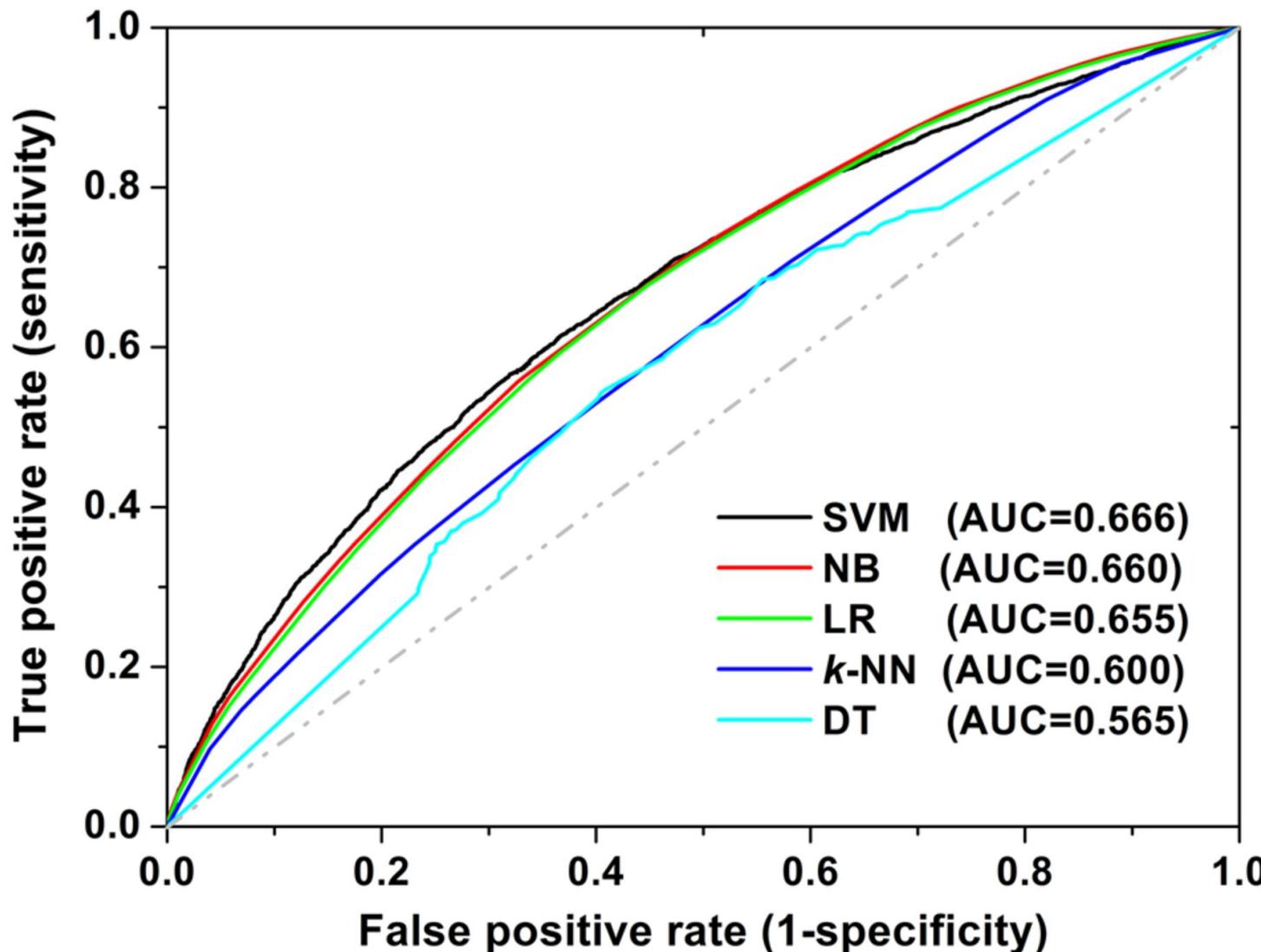
ROC Curves (1/2)

改變 decision boundary 就會改變 sensitivity 與 specificity 的比例



ROC Curves (2/2)

目標在找出 Area Under Curve 最大的線
也就是 sensitivity 和 specificity 同時最好的方法



Game Over

