PYTHON機器學習入門

UNIT 5：CLASSFICATION

授課教師：江尚瑀

SUPERVISED LEARNING- CLASSIFICATION

有許多分類演算法...

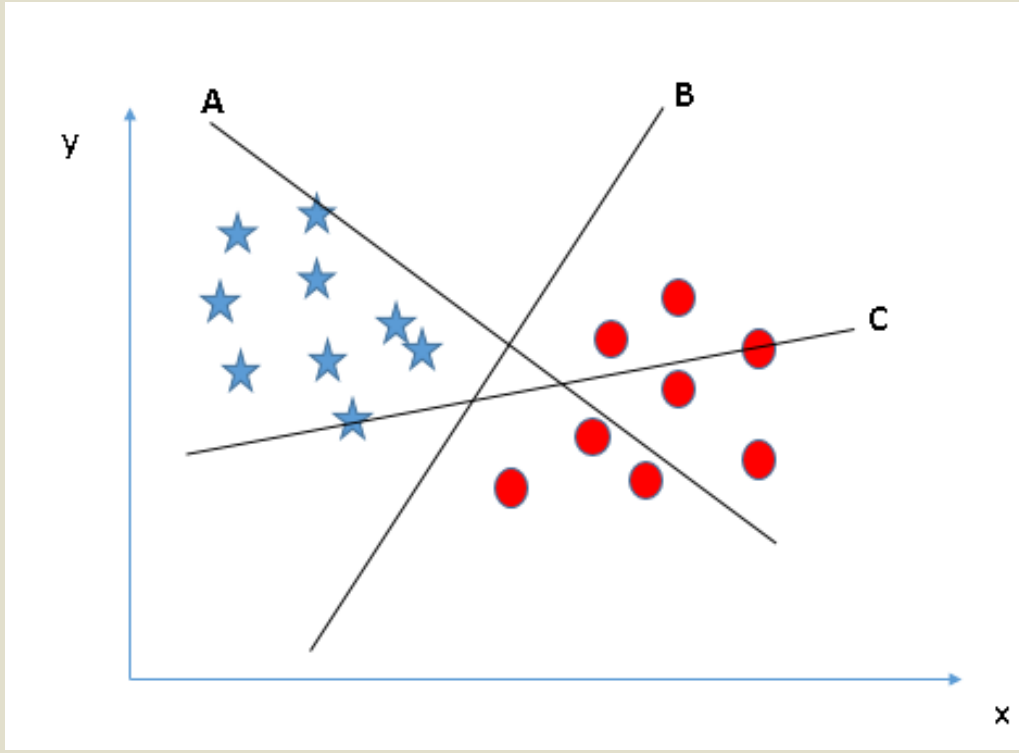
- Logistic Regression
- KNN
- **Support Vector Classifier**
- **Decision Tree**
- **Random Forest**
- ...

SVM
(SUPPORT VECTOR MACHINE)

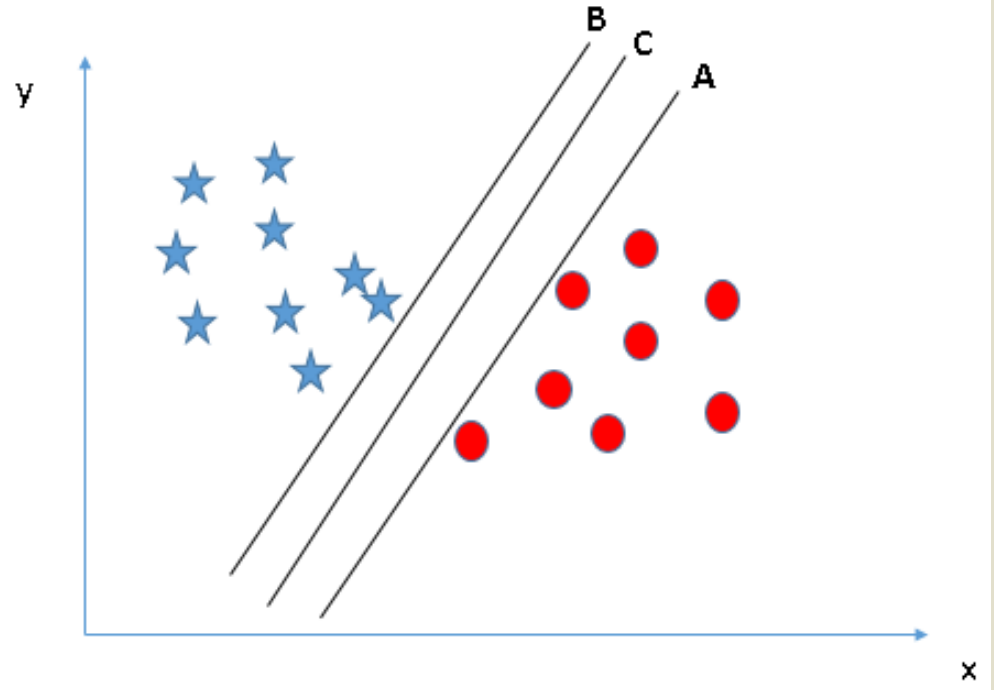
SVM (SUPPORT VECTOR MACHINE)

- a frontier which best segregates the two classes (hyper-plane/ line)

Scenario-1: Identify the right hyper-plane?



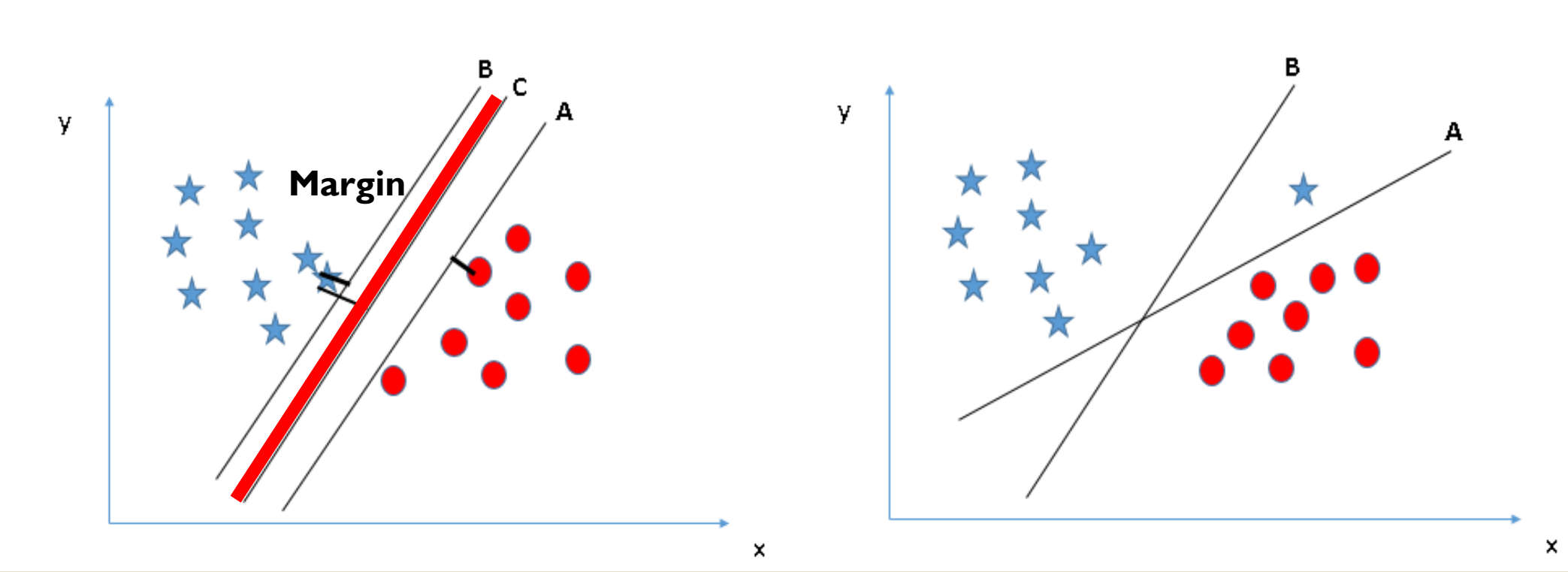
Scenario-2: Identify the right hyper-plane



https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

SVM (SUPPORT VECTOR MACHINE)

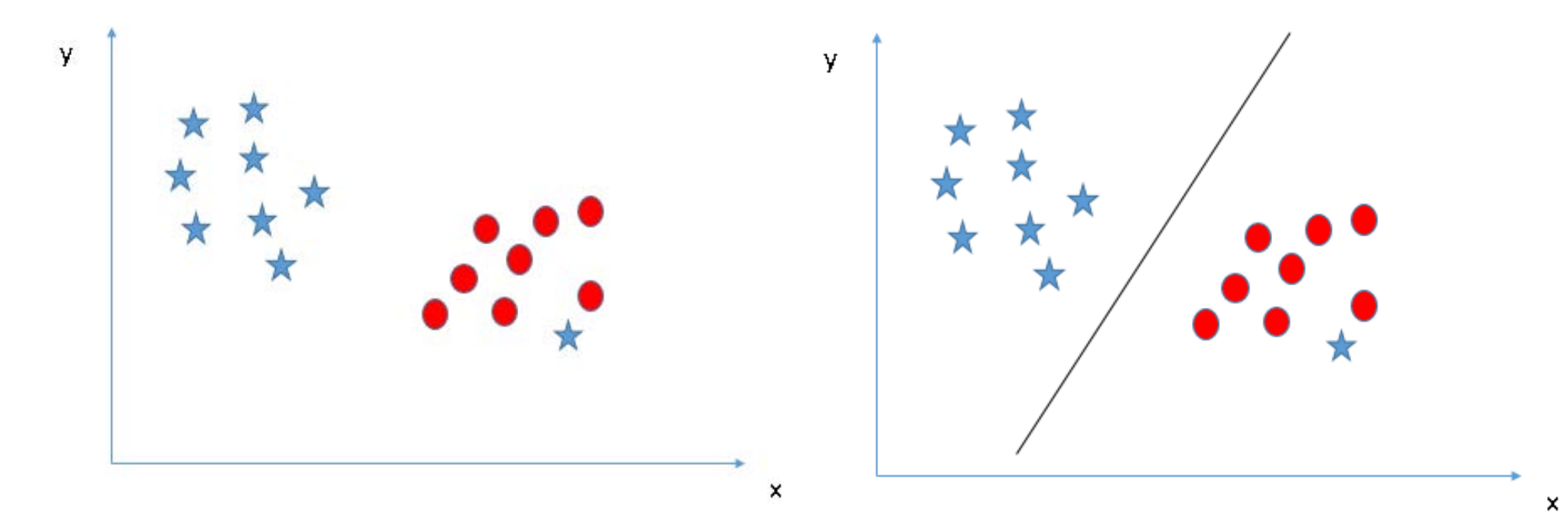
Scenario-3: Identify the right hyper-plane?



maximizing the distances between nearest data point (either class)

SVM (SUPPORT VECTOR MACHINE)

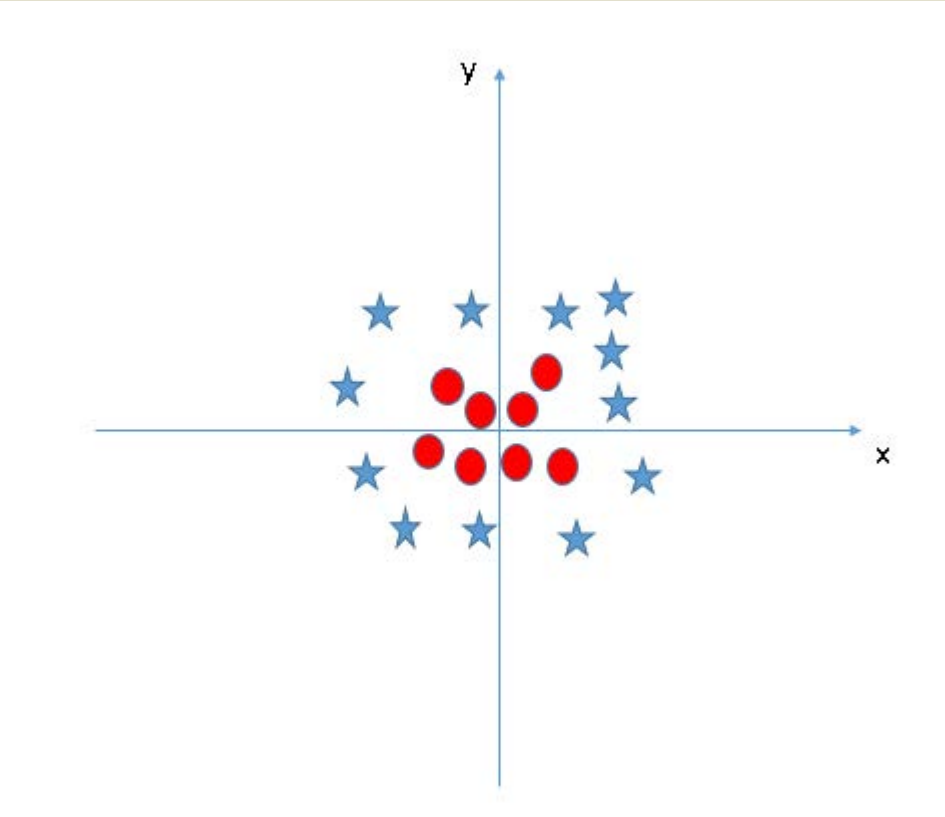
Scenario-4: Can we classify two classes ?



SVM has a feature to ignore outliers and find the hyper-plane that has maximum margin.

SVM (SUPPORT VECTOR MACHINE)

Scenario-5: **Find the hyper-plane to segregate to classes?**

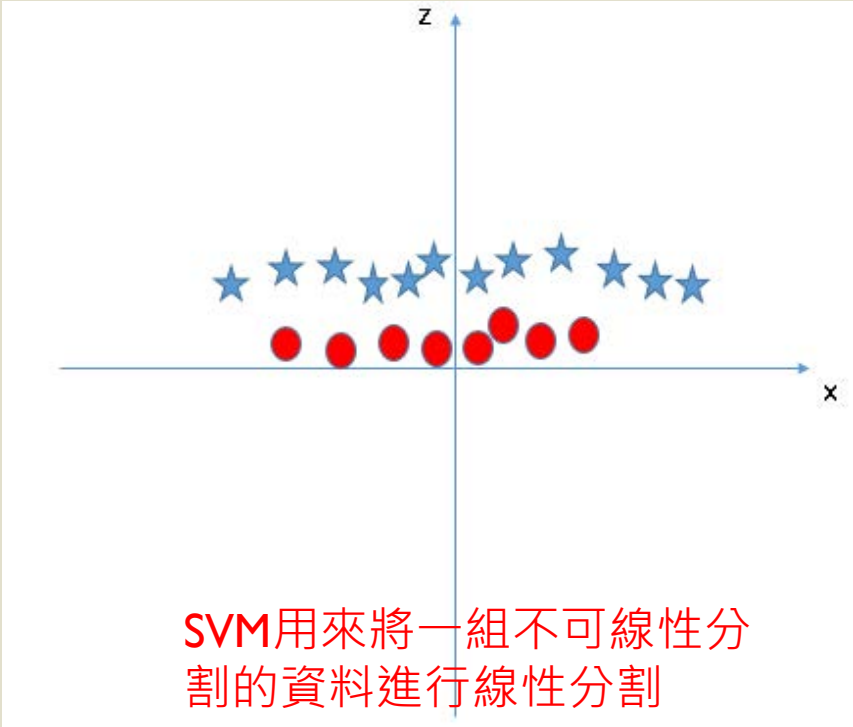


we can't have linear hyper-plane between the two classes

只要對所有的資料，有一個函數可以滿足
k(x,y)=⟨φ(x),ϕ(y)⟩
這個**k(x,y)**就是一個**kernel**函數
⟨**a, b**⟩表示向量**a**和**b**做內積。

KERNEL FUNCTION

SVM solves this problem by introducing additional feature. Here, we will add a new feature z=x2+y2



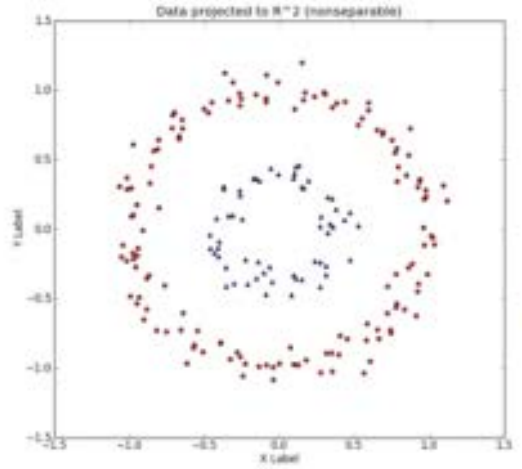
SVM用來將一組不可線性分割的資料進行線性分割

mostly useful in non-linear separation problem

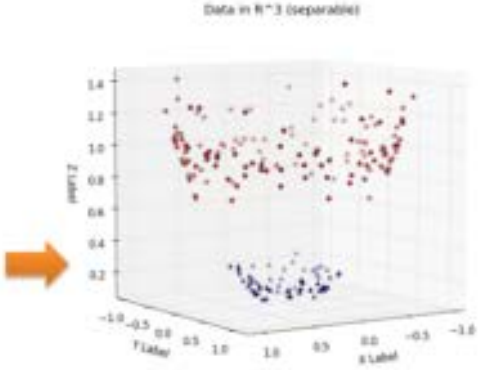
*Should we need to add this feature manually to have a hyper-plane.
No, SVM has a technique called the **kernel trick**.

*These are functions which takes low dimensional input space and transform it to a higher dimensional space i.e. it converts not separable problem to separable problem, these functions are called **kernels**.

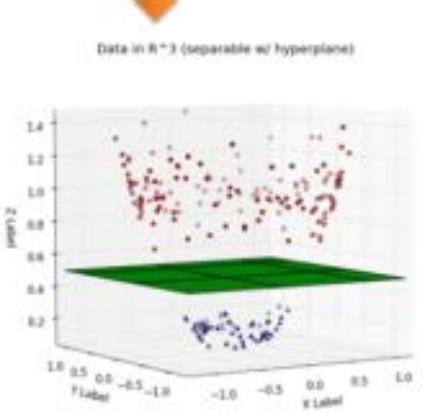
NON-LINEAR FUNCTION



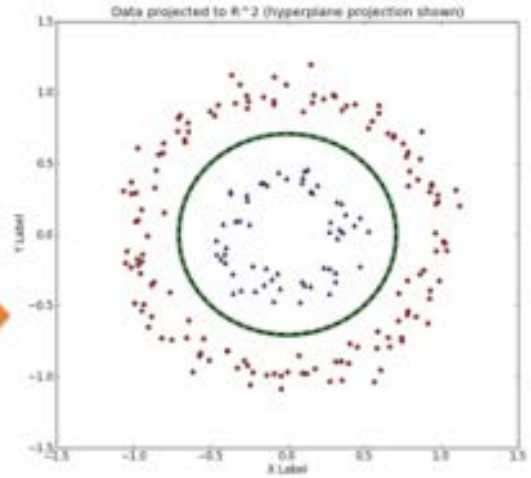
2D平面下為non-linear separable的資料



3D空面可separable



因此可以找到一個hyper plane予以分割



將資料flattern之後轉回2D平面的樣子

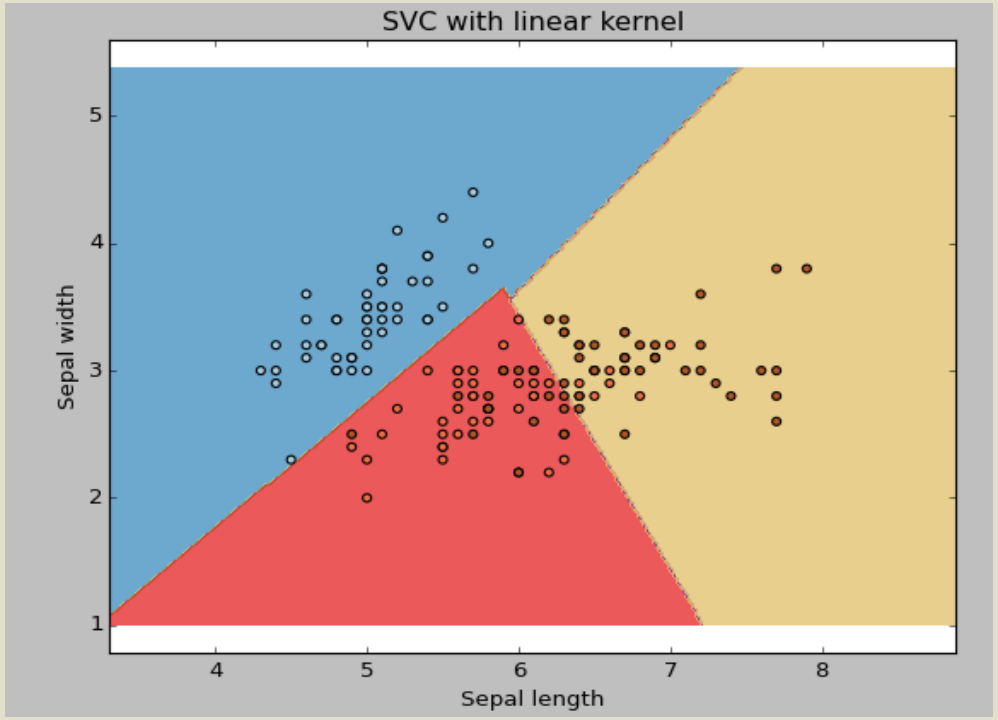
HOW TO TUNE PARAMETERS OF SVM?

- Kernel: ‘linear’, ‘rfb’, ’poly’
- Gamma: Higher the value of gamma, will try to exact fit the as per training data set i.e. generalization error and cause over-fitting problem.
- C: the number of instances that are allowed to fall within the margin, C influences the number of **support vectors** used by the model.

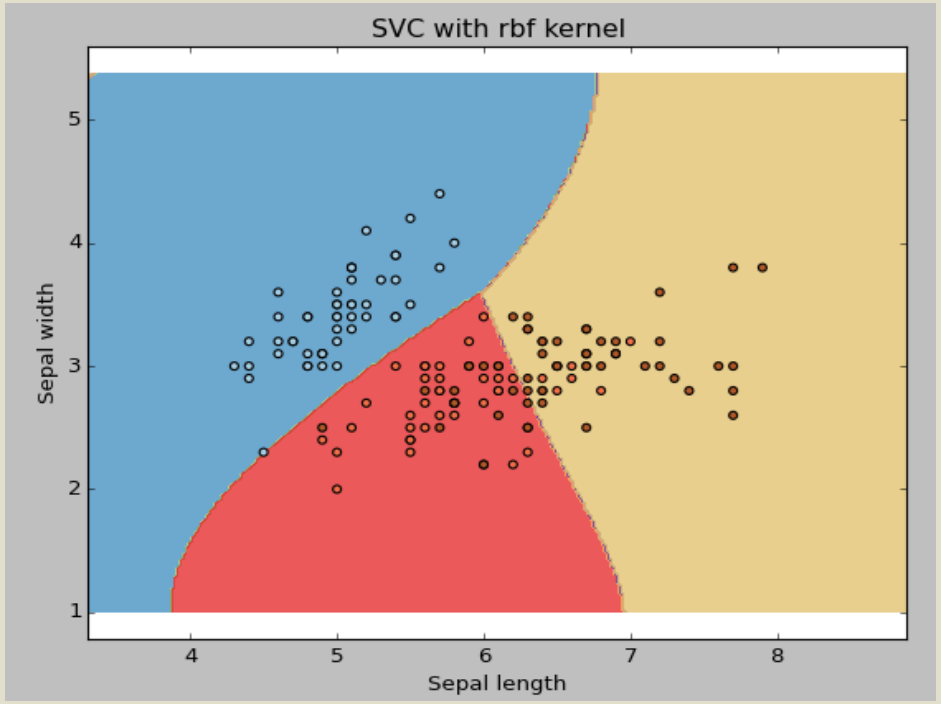
https://www.youtube.com/watch?v=m2a2K4lprQw

HOW TO TUNE PARAMETERS OF SVM?

kernel='linear'



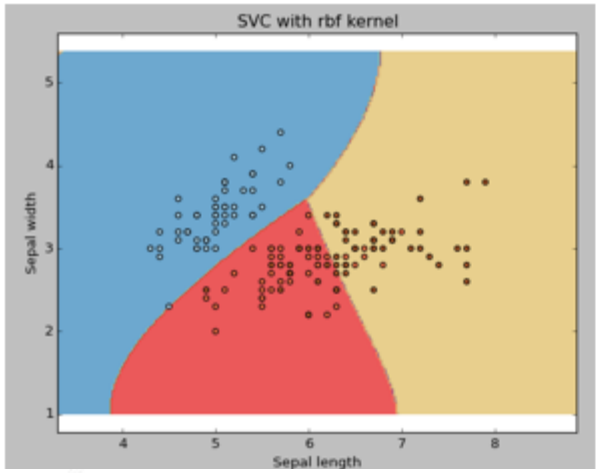
kernel='rbf'



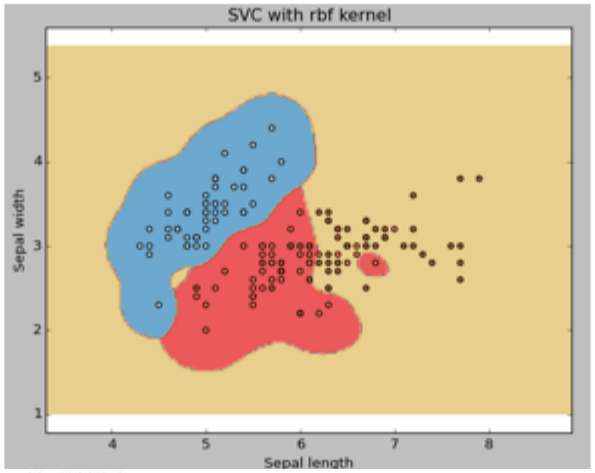
https://cs.stanford.edu/people/karpathy/svmjs/demo/

HOW TO TUNE PARAMETERS OF SVM?

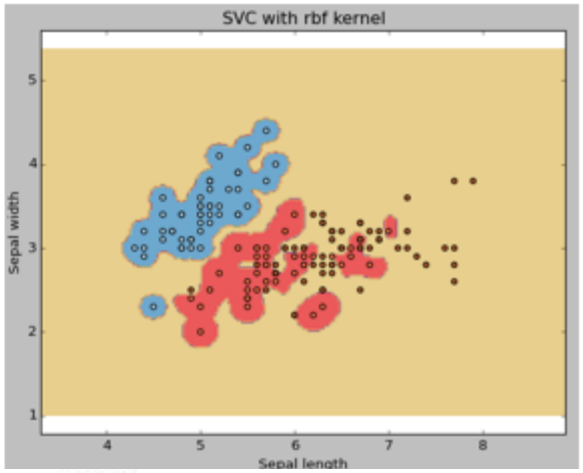
gamma =0



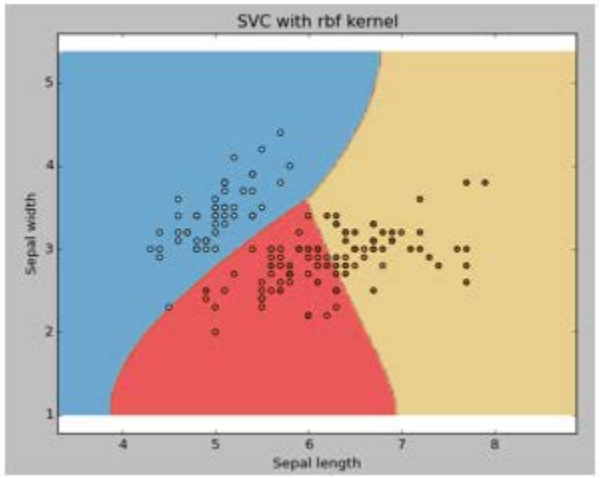
gamma =10



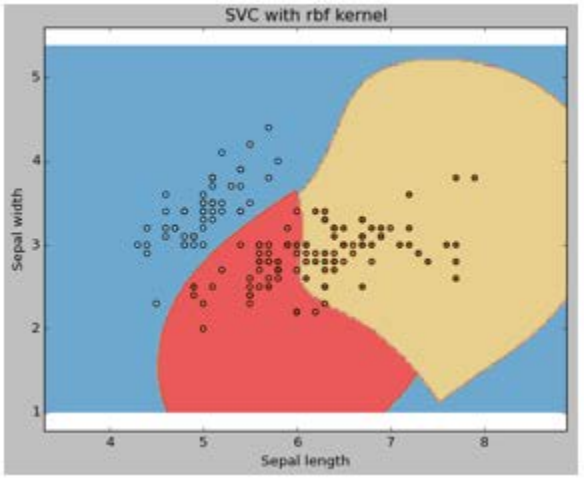
gamma =100



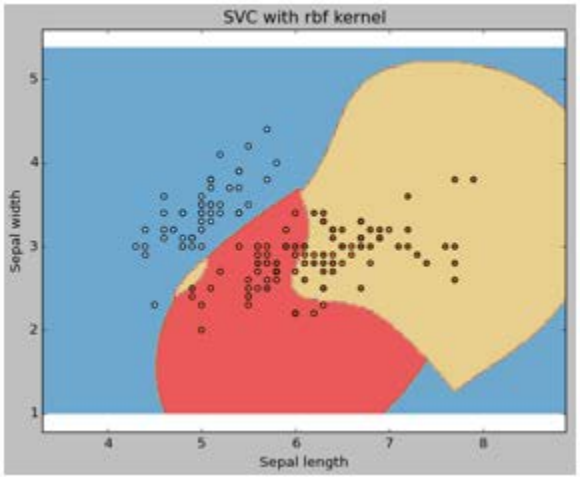
c =1



C =100



c =1000



HOW TO TUNE PARAMETERS OF SVM?

- **gird search**：一種調參手段

```
from sklearn import svm, grid_search

def svc_param_selection(X, y, nfolds):
    Cs = [0.001, 0.01, 0.1, 1, 10]
    gammas = [0.001, 0.01, 0.1, 1]
    param_grid = {'C': Cs, 'gamma' : gammas}
    grid_search = GridSearchCV(svm.SVC(kernel='rbf'), param_grid,
cv=nfolds)
    grid_search.fit(X, y)
    grid_search.best_params_
    return grid_search.best_params_
```

PROS AND CONS ASSOCIATED WITH SVM

- **Pros:**
  - It works really well with clear margin of separation
  - It is effective in high dimensional spaces.
  - It is effective in cases where number of dimensions is greater than the number of samples.
  - It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- **Cons:**
  - It doesn't perform well, when we have large data set because the required training time is higher
  - It also doesn't perform very well, when the data set has more noise i.e. target classes are overlapping
  - SVM doesn't directly provide probability estimates, these are calculated using an expensive five-fold cross-validation. It is related SVC method of Python scikit-learn library.

LAB

- SVM

DECISION TREE

BEFORE MOVE NEXT

舉例來說你是一個披薩公司（像是必X客、達X樂）的資料科學家，你成功建立了一個模型能預測披薩是美味的披薩還是難吃的披薩。利用先前我們介紹的幾種方法，你得到了以下的

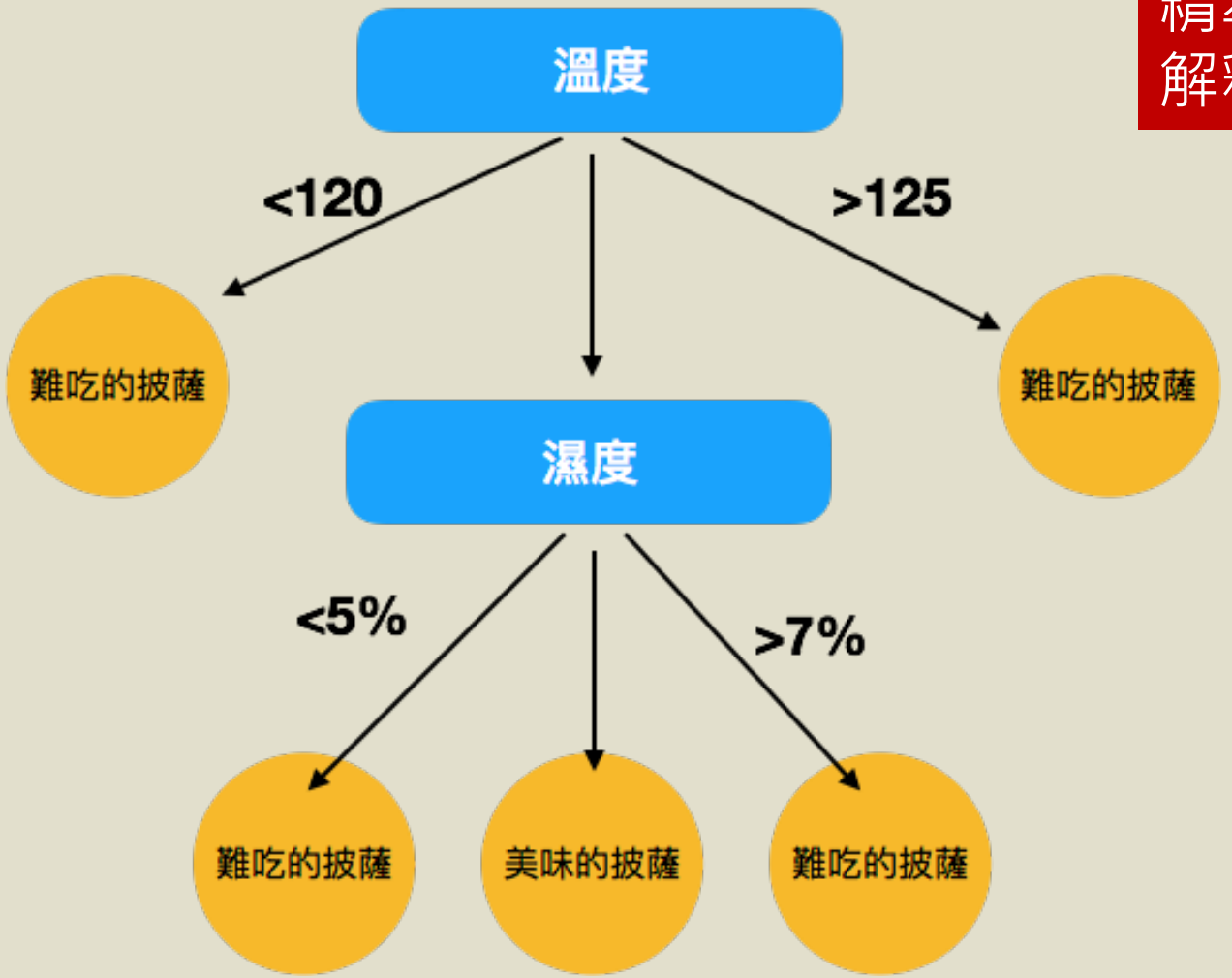
披薩美味方程式: 模型的決策邊界： -100 + 6*溫度 + 3*濕度 = 0

100 + 6*溫度 + 3*濕度 > 0 預測是一個美味的披薩
100 + 6*溫度 + 3*濕度 < 0 預測是一個難吃的披薩

但可能沒人知道你在幹嘛!

決策樹

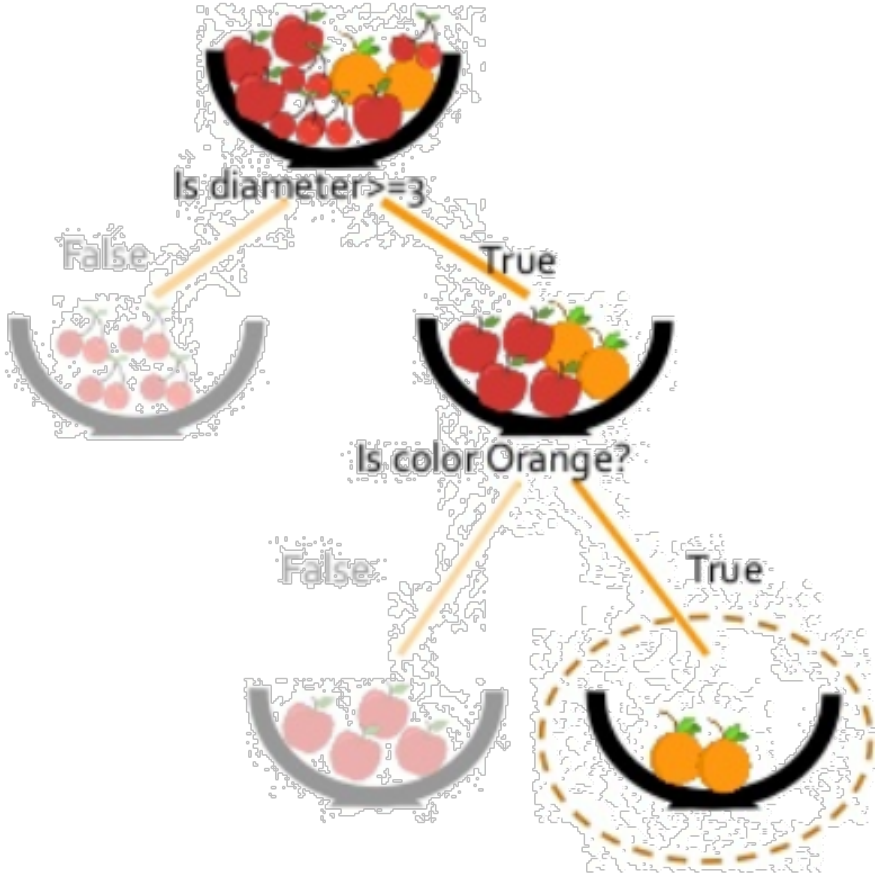
決策樹準確度不一定比較精準，但決策樹模型的可解釋性(Interpretability)高



決策樹(DECISION TREES)

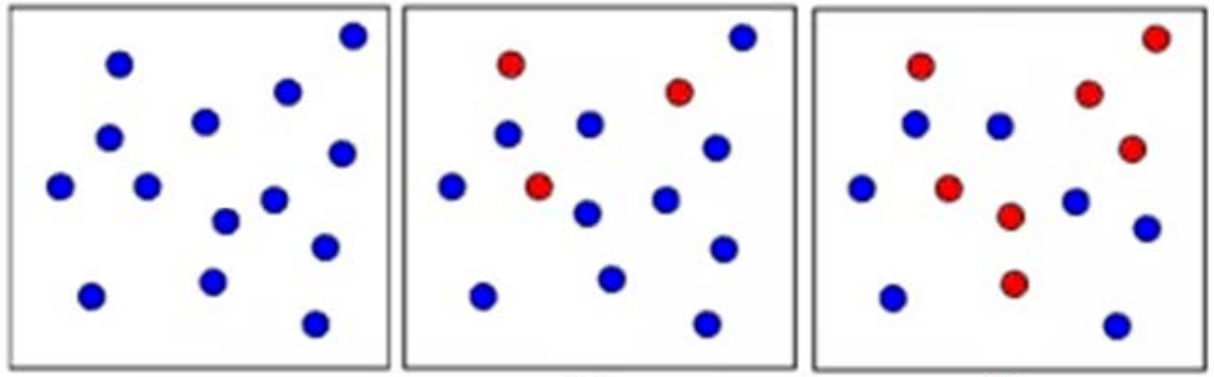
- 是一種條件式的分類器, 專門處理分類問題的樹狀結構

Diameter = 3
Colour = orange
Grows in summer = yes
SHAPE = CIRCLE



如何建構決策樹

- 依據現有資料建構一個決策樹，通常採「由上而下」的方式，從整群依據某個特徵分為數個子群。再從各個子群依據某個特徵，將子群分為更小的子群，直到子群內的資料都是同一個類別為止。
- 問題: 如何在每一步選擇一個最好的特徵來分群? 指使用該特徵時，會使子群中的資料變純



愈純的資料彰顯其意義時，所要使用的資訊量愈少；反之，所使用的資訊量要愈多

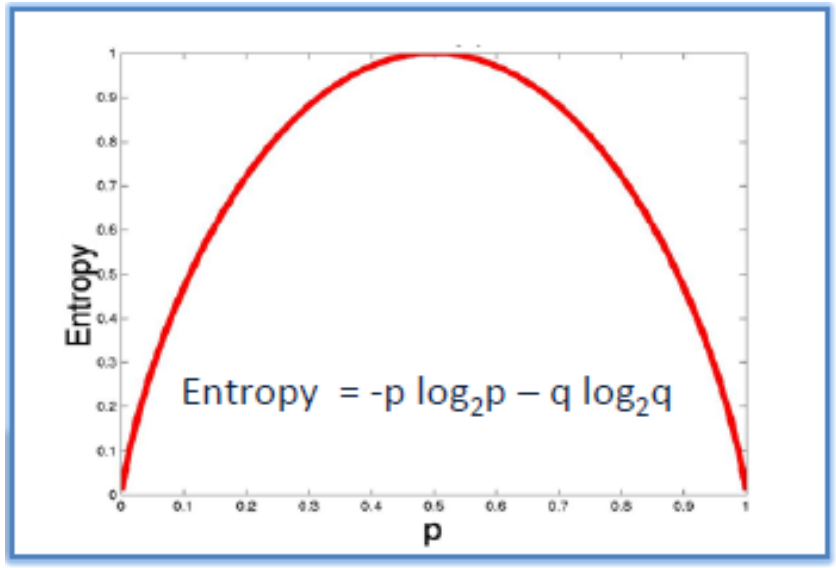
INFORMATION ENTROPY

- Information Entropy : measure of randomness
  More randomness in the data => more entropy

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

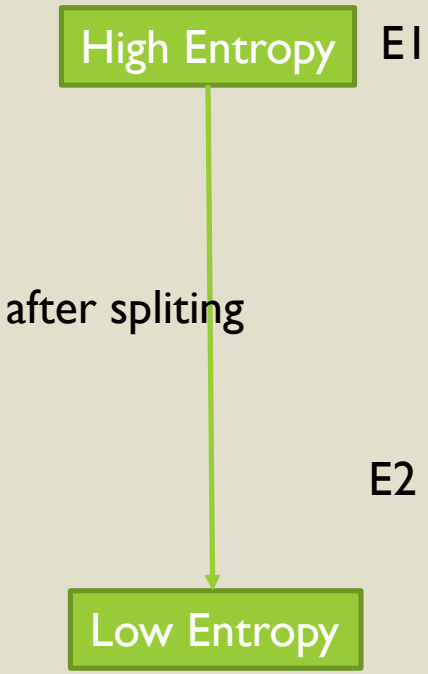
D:代表某一個特徵, 且有m個類別
p: 是某一個類別在這個特徵中出現的機率

若只有兩類時:
當資料全部都是同一類, 則entropy=0;
若資料是各自一半時,entropy=1



Entropy = -0.5 $\log_2$0.5 – 0.5 $\log_2$0.5 = 1

Decision Tree 從**root**到 **leaf**，**entropy** 會由高變低 。因為經過不同的篩子會不斷進行過濾，去蕪存菁 (愈來愈純)。當**entropy=0**時，就是最純的狀態



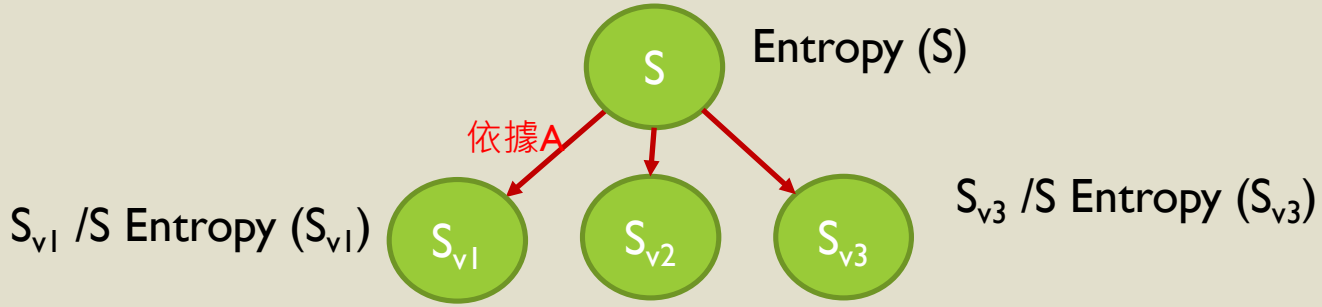
Information gain
=E1-E2

決定那個特徵是最佳分割

- 計算完**entropy**後，接著計算**information gain**
- **Information gain**： 用來衡量某特徵值對於資料分類的能力
- **The information gain, Gain(S, A) of an attribute A, relative to the collection of examples S**

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{S_v}{S} Entropy(S_v)$$

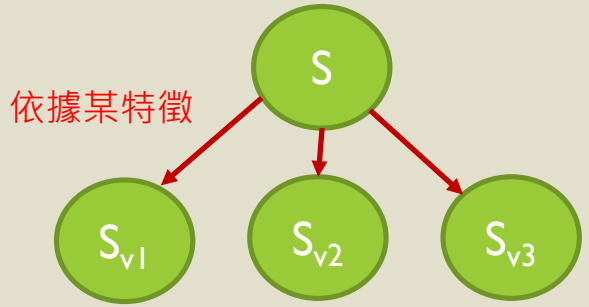
Where Values (A) is the set of all potential values for attribute A, and S_v is the subset of S for which the attribute A has value v.



EXAMPLE

- During two weeks, The target is “play ball?"
- which can be Yes or No.

Outlook	Temp	Humidity	Windy	Play Golf
Rain	Hot	High	false	No
Rain	Hot	High	true	No
overcast	Hot	High	false	Yes
Sunny	Mild	High	false	Yes
Sunny	Cool	normal	false	Yes
Sunny	Cool	normal	true	No
overcast	Cool	normal	true	Yes
Rain	Mild	High	false	No
Rain	Cool	normal	false	Yes
Sunny	Mild	normal	false	Yes
Rain	Mild	normal	true	Yes
overcast	Mild	High	true	Yes
overcast	Hot	normal	false	Yes
Sunny	Mild	High	true	NO



Step 1: Calculate entropy of Play Golf (target).

Play Golf	
Yes	No
9	5

Entropy(Play Golf) = Entropy (5,9)
= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)
= 0.94

EXAMPLE

Step 2: The dataset is then split into the different attributes.

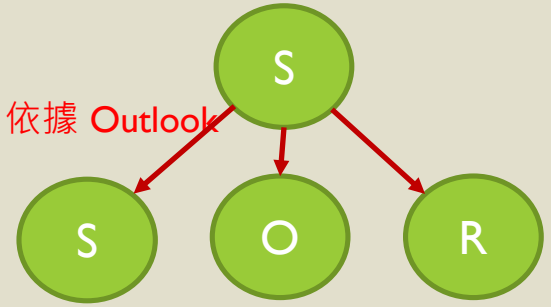
		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain=0.247			

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain=0.029			

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain=0.152			

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain=0.048			

IG 愈大表示此特徵內資料凌亂程度愈小



Outlook Gain: 0.94- (5/14*E(3,2)+4/14*E(4,0)+5/14*E(2,3))=0.247

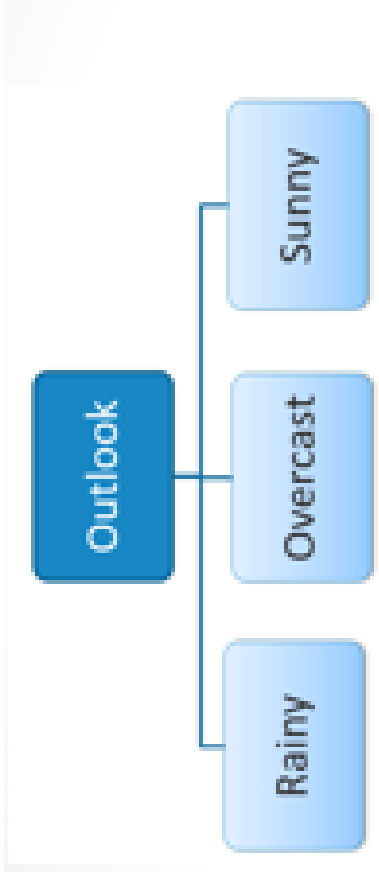
Temp Gain: 0.94- (4/14*E(2,2)+6/14*E(4,2)+4/14*E(3,1))=0.029

Humility Gain: 0.94- (7/14*E(3,4)+7/14*E(6,1))=0.152

Windy Gain: 0.94- (8/14*E(6,2)+6/14*E(3,3))=0.048

Step 3: pick out attribute with the greatest IG as the decision node

EXAMPLE

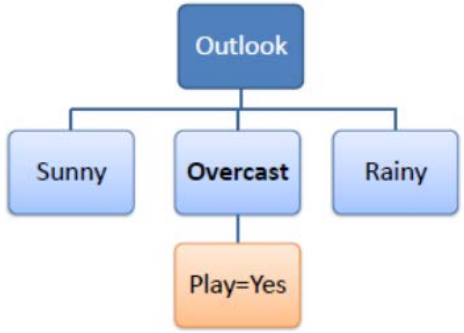


Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

EXAMPLE

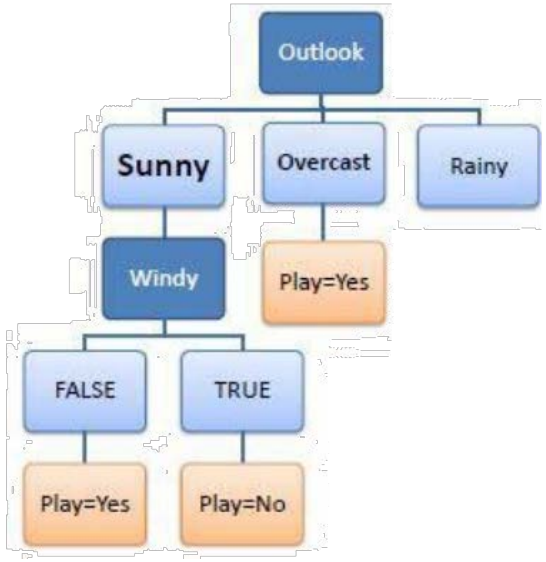
Step 4a: A branch with the entropy of Outlook= overcast.

Temp	Humidity	Windy	Play Golf
hot	high	False	Yes
cool	normal	True	Yes
mild	high	True	Yes
hot	normal	False	Yes



Step 4b: A branch with entropy of Outlook= sunny (Windy=False & Windy=True).

Temp	Humidity	Windy	Play Golf
mild	high	False	Yes
cool	normal	False	Yes
mild	normal	False	Yes
mild	high	True	NO
cool	normal	True	No



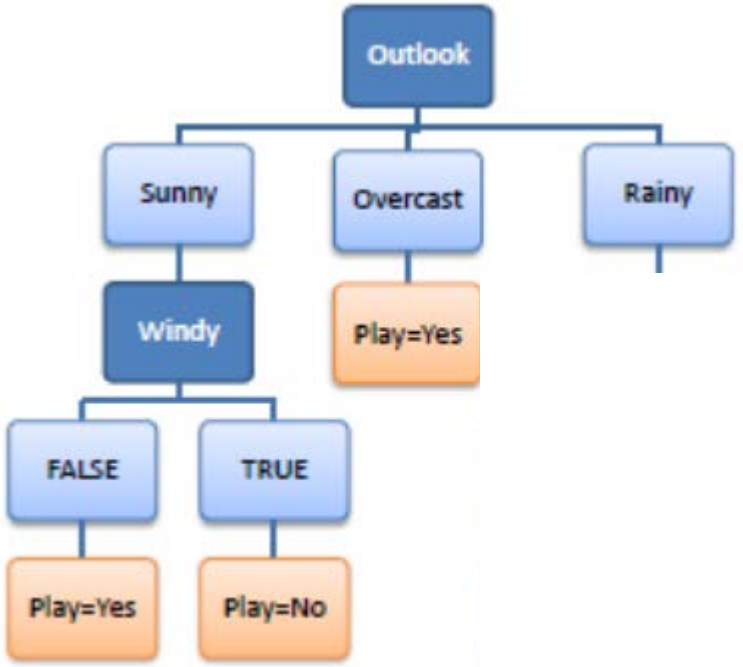
EXAMPLE

Step 4c: A branch with entropy of Outlook= rain (Humidity= high & Humidity = normal)

Temp	Humidity	Windy	Play Golf
hot	high	false	No
hot	high	true	No
mild	high	false	No
Cool	normal	false	Yes
mild	normal	true	Yes

Exercise :

試著計算Temp, Humidity, Windy 的 Information Gain
並畫出在Rainy 下的樹狀結構



Decision Tree of Weather data sets

決策樹優缺點

Pros:

- Implicitly perform feature selection
- Easy to interpret and explain.
- Can generate rules helping experts to formalize their knowledge.
- Data classification without much calculations
- Handling both continuous and discrete data
- Require relatively little effort from users for data preparation
  - they do not need variable scaling;
  - they can deal with a reasonable amount of missing values;
  - they are not affected by outliers.

決策樹優缺點

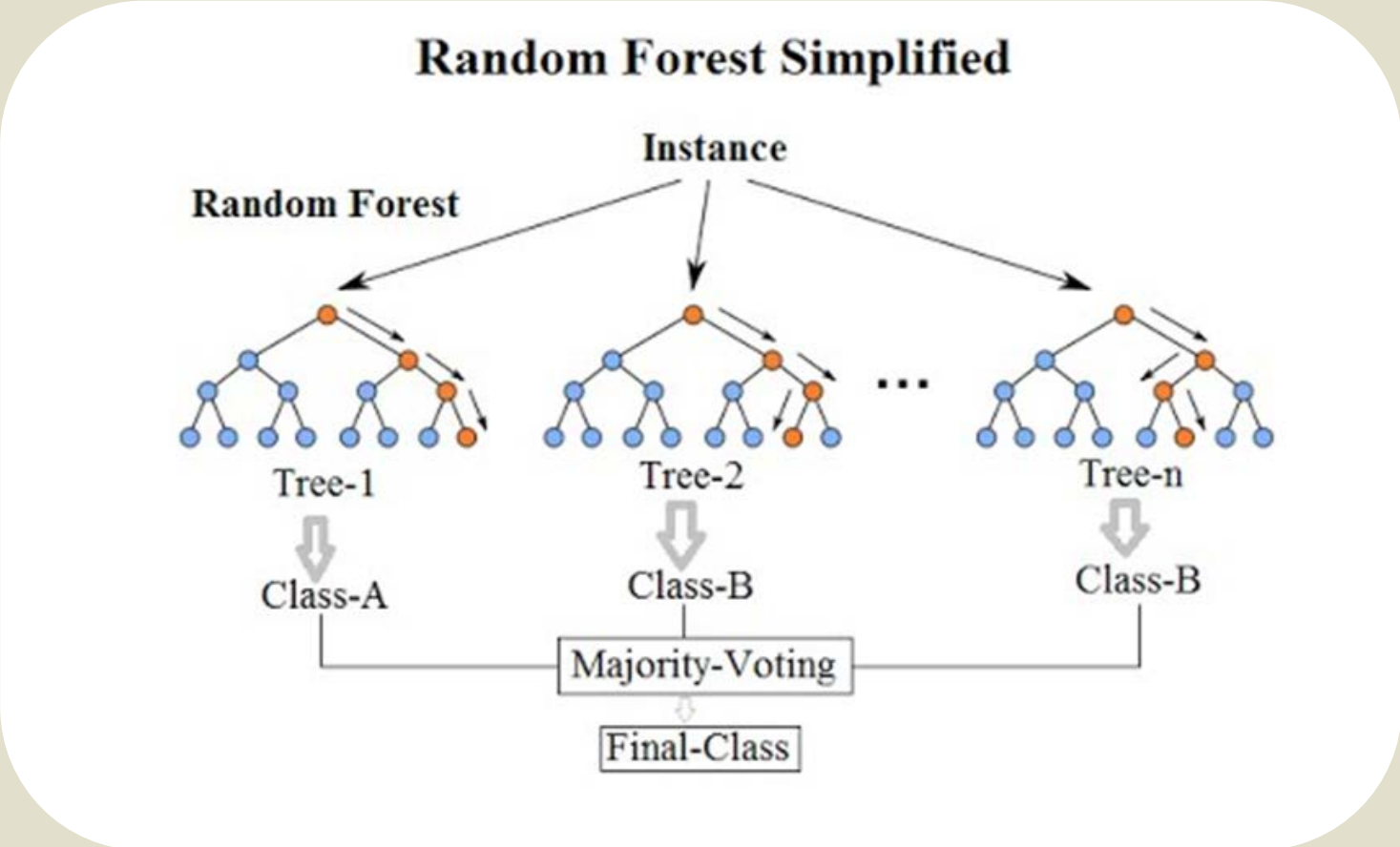
Cons:

- **The high classification error rate while training set is small in comparison with the number of classes**
- **Exponential calculation growth while problem is getting bigger.**

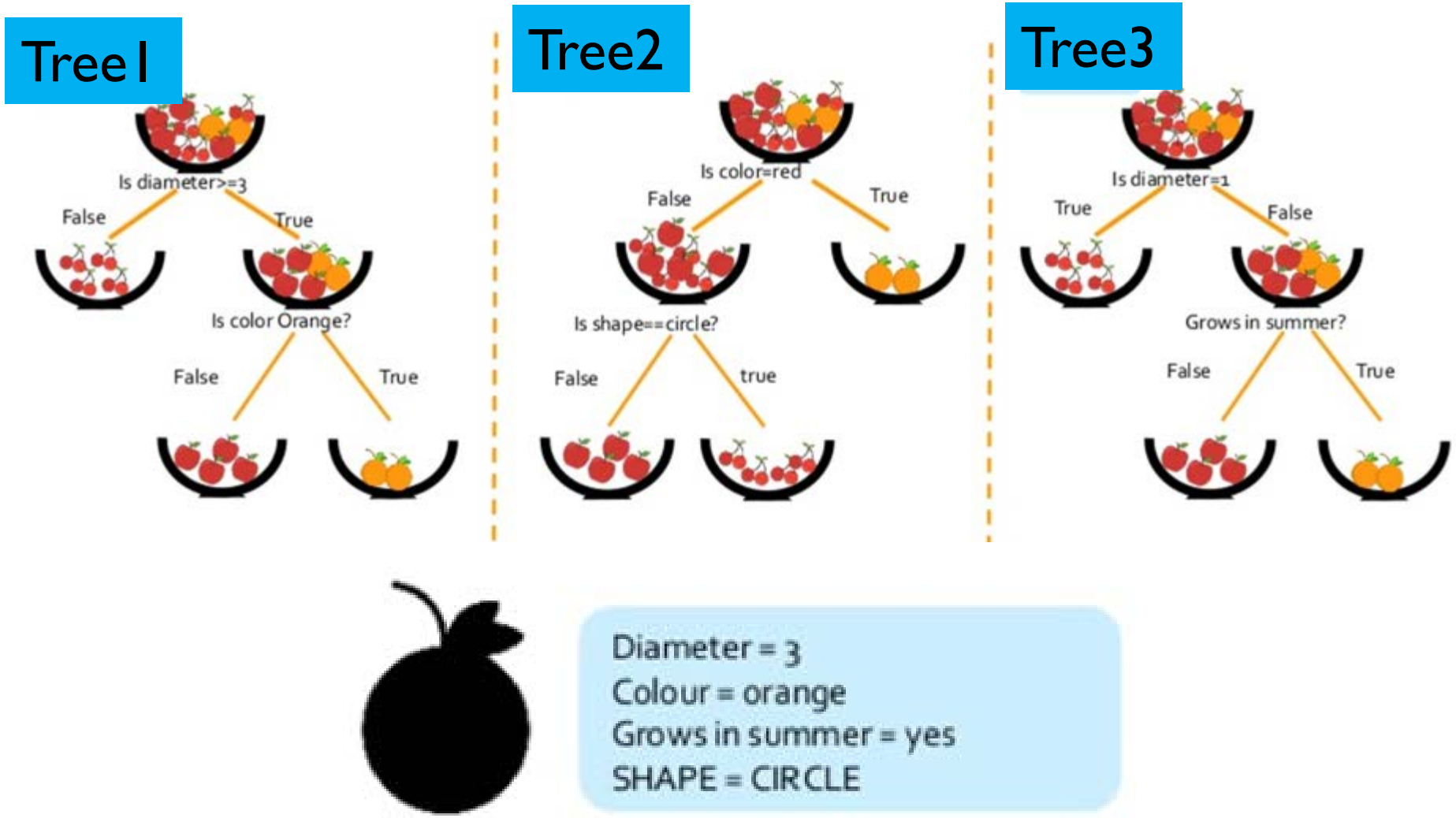
RANDOM FOREST

隨機森林(RANDOM FORESTS)

- 在訓練資料中, 取部份特徵與部份資料產生樹。每重複此步驟, 就會再產生一棵樹
- 多棵樹採 **Ensemble (Majority Vote)** 的方式決定最終判斷



HOW DOES IT WORK?



HOW DOES IT WORK?

Diameter = 3
Colour = orange
Grows in summer = yes
SHAPE = CIRCLE

Tree1:

orange

Tree2:

cherry

Tree3:

orange

Majority Vote

orange