高度不平衡數據的處理與建議

金融數位轉型Workshop-個金行銷實務

弘智科技 洪瑞隆

個人簡介

- 終身教授, Boise State University
- 大數據研究中心 (德州理工大學進階分析與商業智慧中心、中佛羅裡達大學大數據研究中心)
- The Asian Banker 最佳資料分析大獎、最佳雲運用大獎
- 弘智科技技術長
- 大數據顧問: 銀行業、資策會、高鐵
- 多項分析相關發明與新型專利
- 機器學習與深度學習在金融業應用

課程假設

- 各位均有基礎的機器學習的概念或建模經驗
- 課程的重點在於不平衡數據的處理策略、策略比較與建議
- 各模型基本精神會大略介紹，但不會花太多時間優化個別算法

機器學習的定義

- 什麼是"學習"？學習就是人類通過觀察、積累經驗，掌握某項技能或能力。而機器學習（Machine Learning），顧名思義，就是讓機器（電腦）也能向人類一樣，通過觀察大量的資料和訓練，發現事物規律，獲得某種分析問題、解決問題的能力。
- 機器學習可以被定義為：機器從資料中總結經驗，從資料中找出某種規律或者模型，並用它來解決實際問題。

機器學習範例

- 輸入: 客戶申請資料
- 輸出: 核卡後是否發生違約
- 目標函數: 理想的信用卡核卡公式
- 數據: 歷史資料
- 假設、建模知識、機器學習算法 (數據轉換、損失函數、衡量指標) => 找出最貼近目標函數的公式

解釋模型與預測模型

解釋模型

- 主要重點在因子
- X & Y 之間的關係
- 為了描述
- 小樣本
- 變量少
- 使用P值與信賴區間

預測模型

- 主要重點在Y
- 為了預測
- 大樣本
- 變量多
- 使用驗證資料來評估模型

機器學習類型

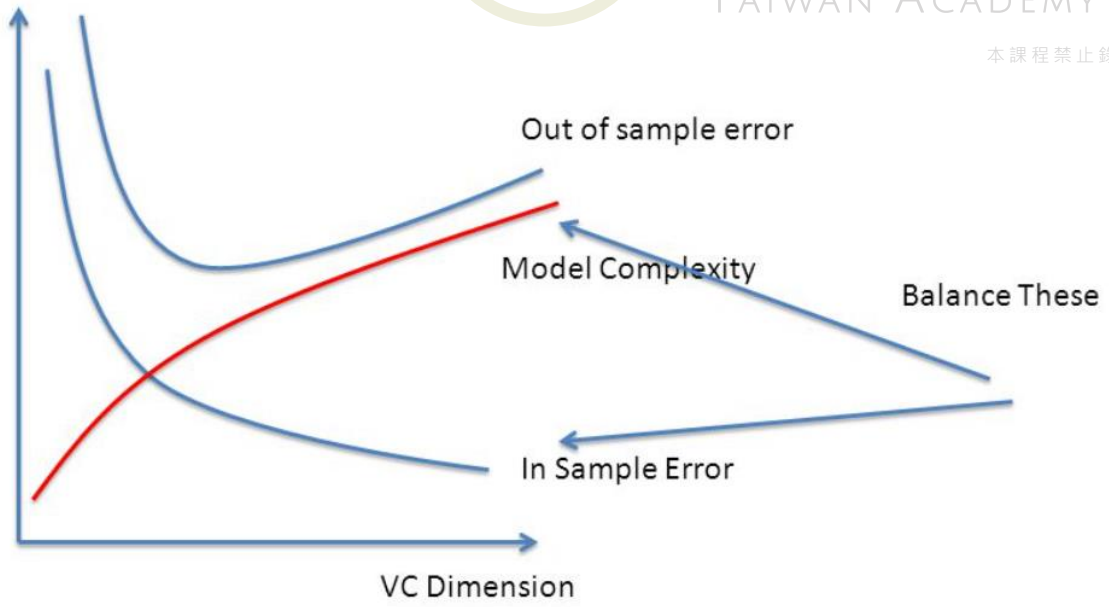
- 非監督式學習:非監督式學習是沒有輸出標籤yn的 (目標變量)，典型的非監督式學習包括：聚類（Clustering）問題，比如對網頁上新聞的自動分類；密度估計，比如交通路況分析與異常檢測，
- 監督式學習:如果我們拿到的訓練樣本D既有輸入特徵 x，也有輸出 yn，那麼我們把這種類型的學習稱為監督式學習（Supervised Learning）。監督式學習可以是二元分類、多元分類或者是回歸 (數值型目標)
- 半監督式學習:半監督式學習就是說一部分資料有輸出標籤 yn，而另一部分資料沒有輸出標籤 yn ，目前實務中有很多僅標註一部分資料，大部分都是沒標註的
- 增強學習: 在現實生活中，給模型或系統一些輸入，但是給不了我們希望的真實的輸出y，根據模型的輸出回饋，如果回饋結果良好，更接近真實輸出，就給其正向激勵，如果回饋結果不好，偏離真實輸出，就給其反向激勵。不斷通過"回饋-修正"這種形式，一步一步讓模型學習的更好，如高鐵的區間座位調控，下棋等。
- 生成模型: 在概率統計理論中, 生成模型是指能夠隨機生成觀測數據的模型，尤其是在給定某些隱含參數的條件下。 它給觀測值和標註數據序列指定一個聯合概率分布。 在機器學習中，生成模型可以用來直接對數據建模（例如根據某個變量的概率密度函數進行數據採樣），也可以用來建立變量間的條件概率分布。 條件概率分布可以由生成模型根據貝葉斯定理形成

泛化能力、準確度與模型複雜度之關係

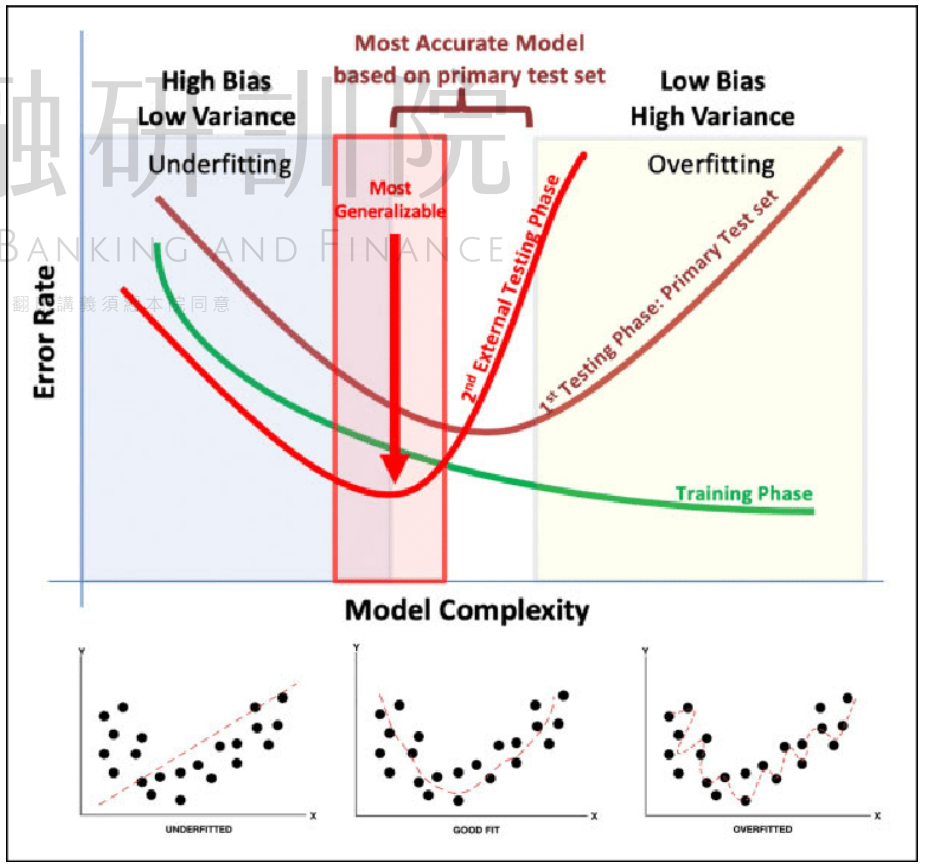
圖片出處

$$E_{out}(h) \leq E_{in}(h) + \Omega(N, \mathcal{H}, \delta)$$

In Sample Error + Model Complexity



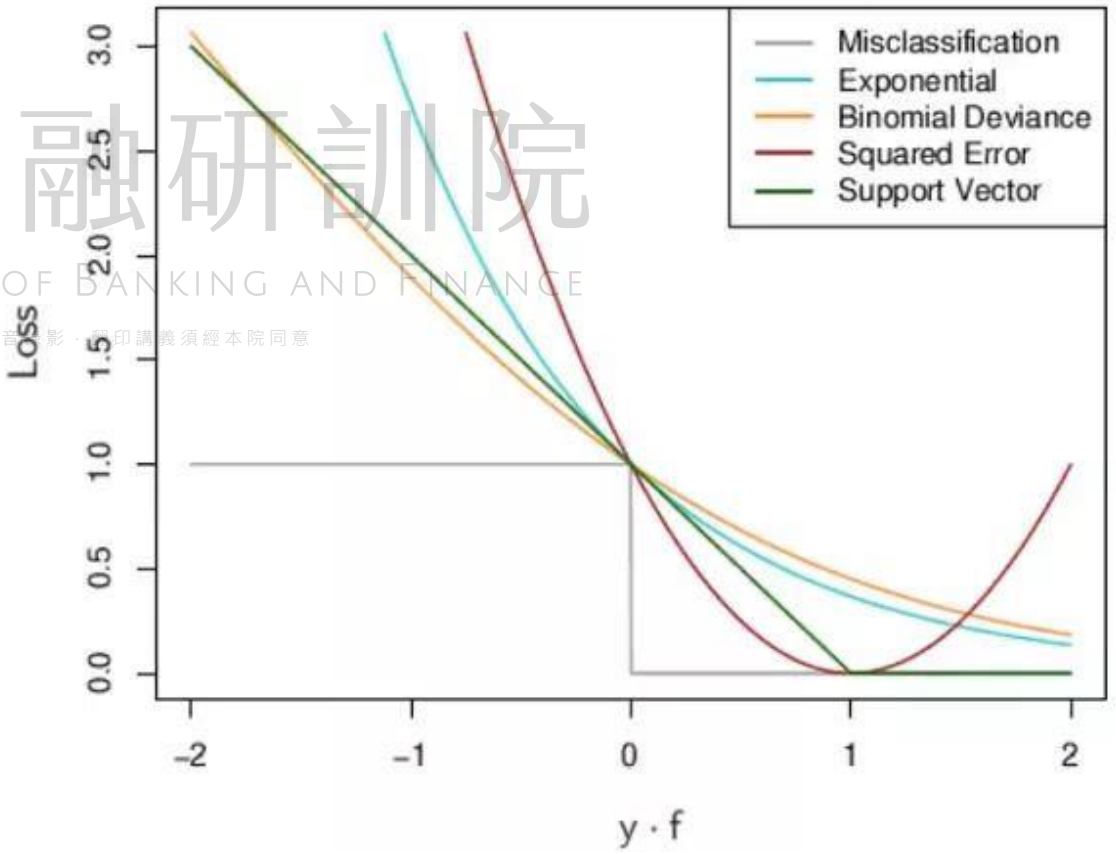
圖片出處



算法就是數據轉換與損失函數搭配

圖片出處

- 數據轉換: Domain knowledge
- 轉換函數: VC 特性
- 損失函數:
  (如何評估該模型最接近完美狀態)



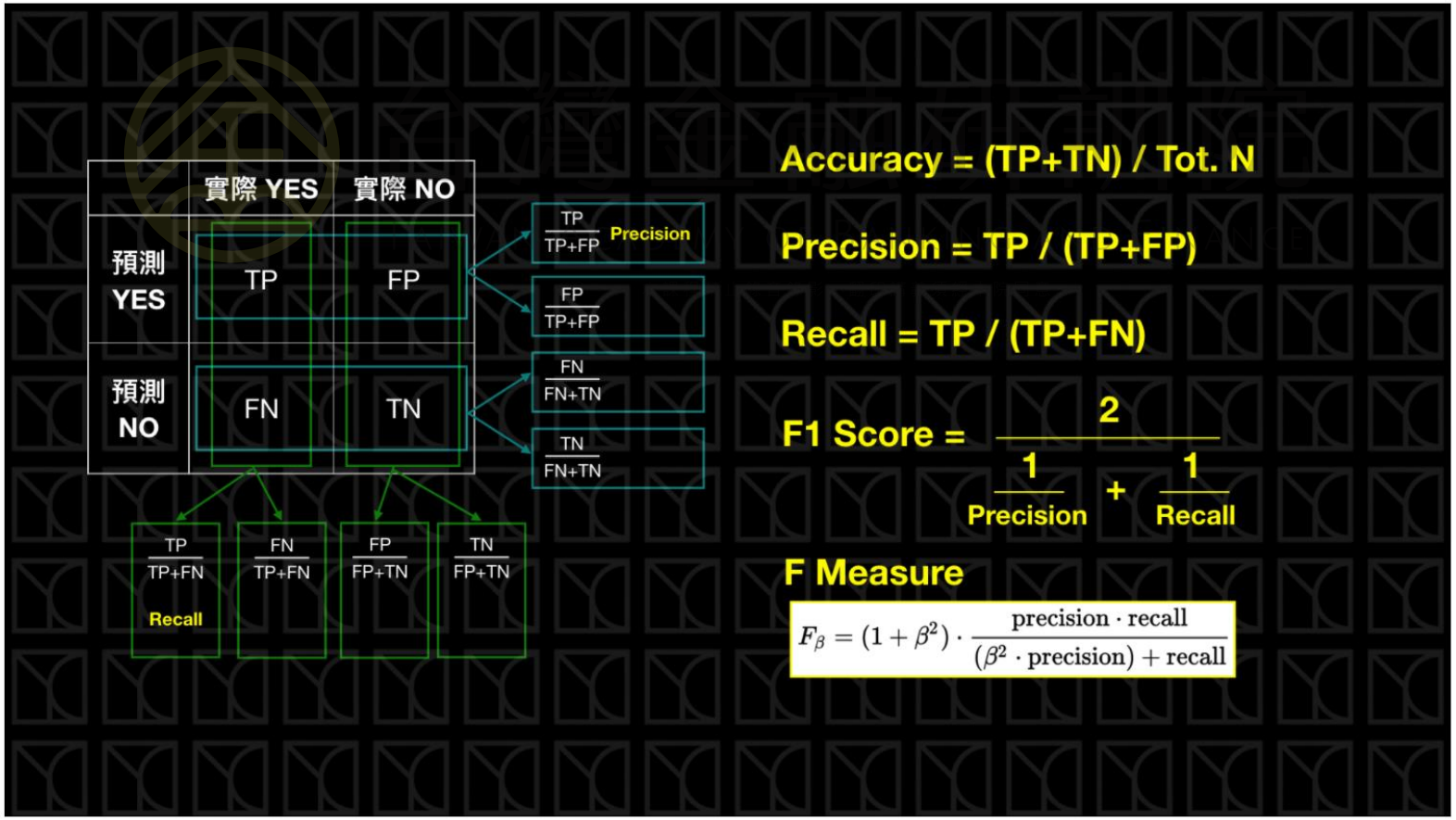
Confusion Matrix (混淆矩陣)

True/False 預測正確？ **Positive/Negative** 預測方向

	實際 YES	實際 NO
預測 YES	TP (True Positive)	FP (False Positive) Type I Error
預測 NO	FN (False Negative) Type II Error	TN (True Negative)

常用模型指標

- Accuracy 在高度不平衡數據不一定重要
- Recall 是最重要指標
- Precision 其次



練習

	實際 Yes	實際 No
預測 Yes	20	30
預測 No	50	900

- Sample size:?
- 全部違約 cases:?
- 全部正常 cases:?
- 真實違約率:?
- Accuracy: ?
- Recall: ?
- Precision_0: ?
- Precision_1: ?

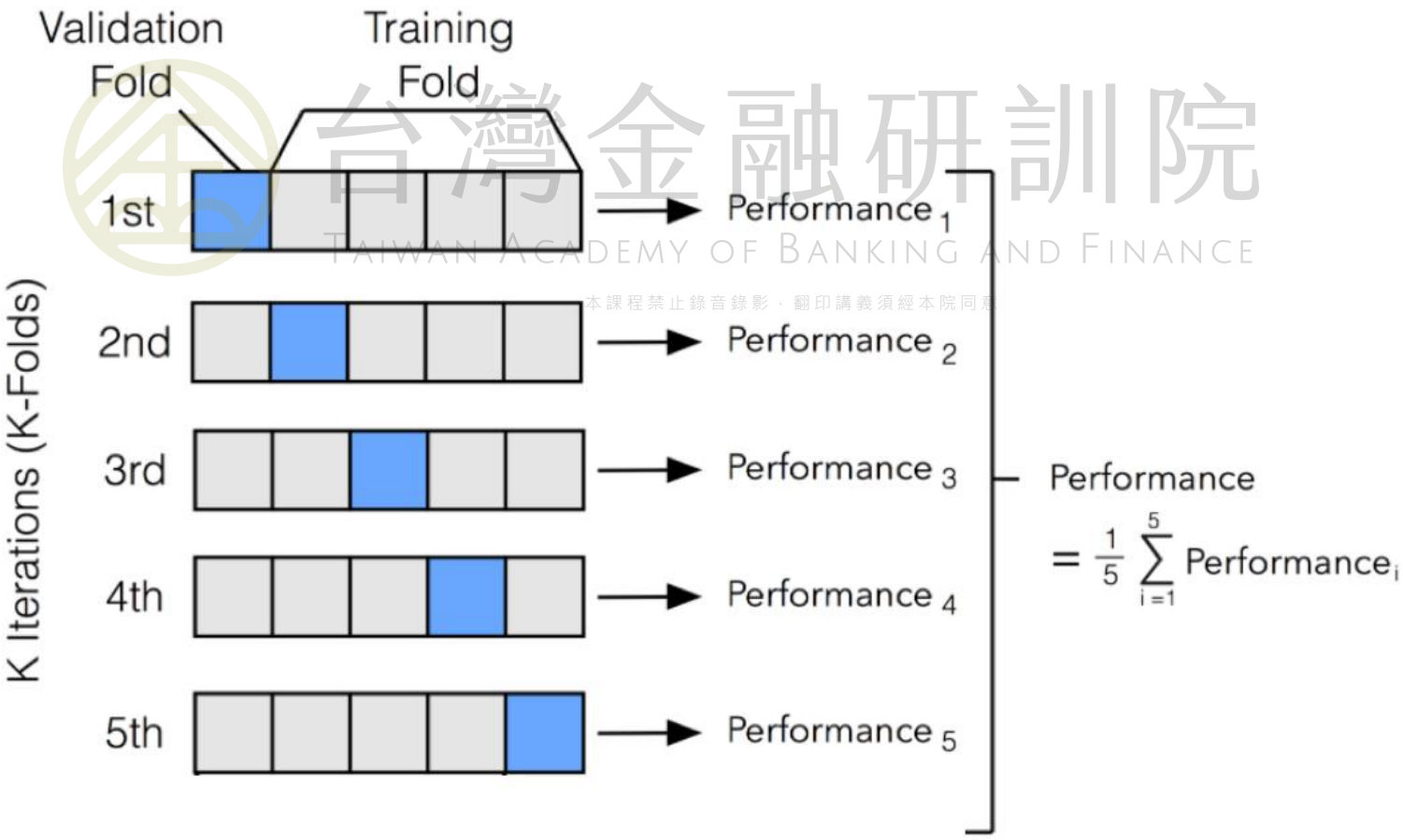
模型編碼有關係嗎?

- 風險模型，違約的人應編碼為?
- 行銷模型，會購買的人應編碼為?

哪些指標比較重要?

- True Positive和True Negative 越大越好
- False Positive 和 False Negative 越小越好
- 風險模型 => False Negative (違約為 1) 越小越好
- 風險模型 => False Positive (白名單為 1) 越小越好
- 行銷模型 => False Negative (回應為 1) 越小越好

Cross Validation (交叉驗證)



不平衡數據處理

何謂不平衡數據

- Imbalanced data: 目標樣本佔所有樣本的比例較懸殊 (5%以下)
- 在模型的建置上需要特殊的技巧

不平衡數據可能產生的問題

- 預測不準確或是模型表現不佳
  - 模型無法達到預期的效果 (e.g. 3% 忽略 => 97% 準確) => 高準確但無用的模型
  - 為了抓到目標，須包含很多假訊號
    (降低閥值來提高捕抓率 => 降低模型訊號準確度)
  - 樣本數少到無法對於重要特徵做辨認

不平衡數據建模方法分類

傳統建模方法

- Threshold adjustment
- 權重法
- Oversampling
- Undersampling
- SMOTE

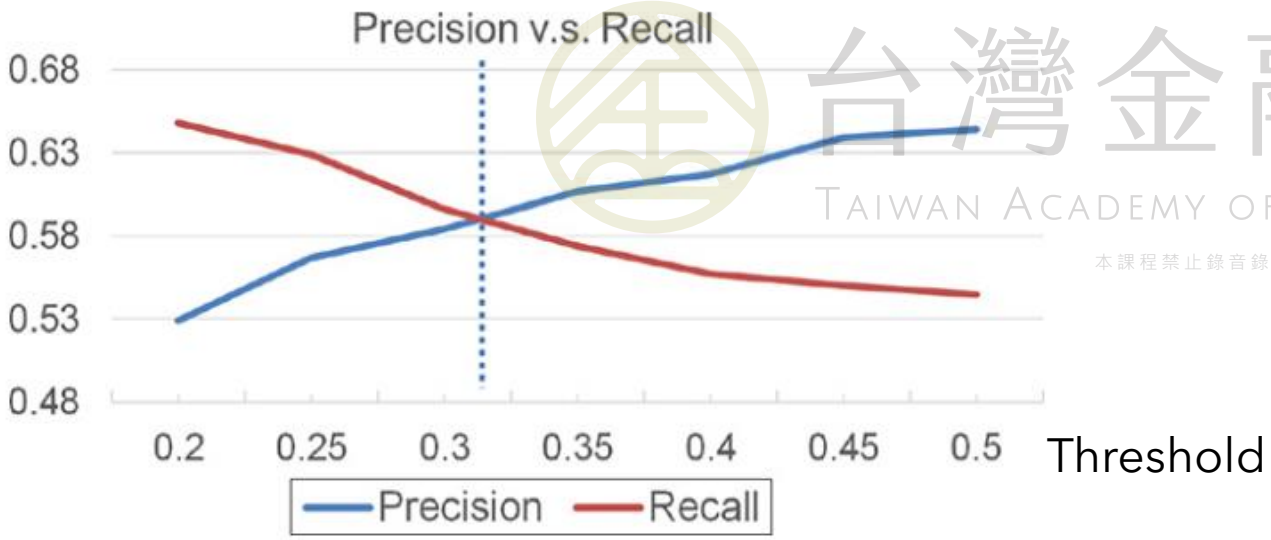
深度學習方法

- VAE Autoencoder
- Others:
  - GAN
  - Transfer learning or self-constructed models (其他技術手段)

閥值調整法 (Threshold Adjustment)

Subtitle

Threshold adjustment



- 風險模型: 違約或是白名單 (Recall, Precision, & F1)
- 行銷模型: Recall & Accuracy
- 閥值決定: 原始 (0.5)，但實務可彈性決定
- 調高閥值 => Recall 變低；Precision 變高
- 降低閥值 => Recall 變高；Precision 變低
- 平衡點：Recall & Precision 交叉點

權重法

- 透過增加和減少少數類和多數類的權重。如果給少數類賦予非常高的類權重，演算法很可能會偏向少數類，並且會增加多數類中的錯誤。

Undersampling & Oversampling

- Oversampling: 加大目標 (增加 Minority 數量; i.e. 1)
- Undersampling: 減少 0 (減少 Majority 數量; i.e. 0)
- Imbalanced 比例: 1:3

Undersampling & Oversampling 的影響

- 機率數值被改變，但是機率順序不變
- 參數不變但是截距變了
- False negative & False positive rates 變了，但是這個跟閥值相關 (可考慮調整閥值)
- ROC curve 不變

SMOTE

- Oversampling: 用 K-NN 產生假樣本
- Undersampling: 用 K-NN 刪除 0

- SMOTE 介紹

Q &A

在此次課程中使用的分類模型

- 羅吉斯迴歸 (Logistic Regression)
- 決策樹 (Decision Tree)
- K-NN 近鄰分類法 (K Nearest Neighbors)
- 隨機森林 (Random Forest)
- 支持向量機 (Support Vector Machine)

Q &A

迴歸分析 (Regression)

Subtitle

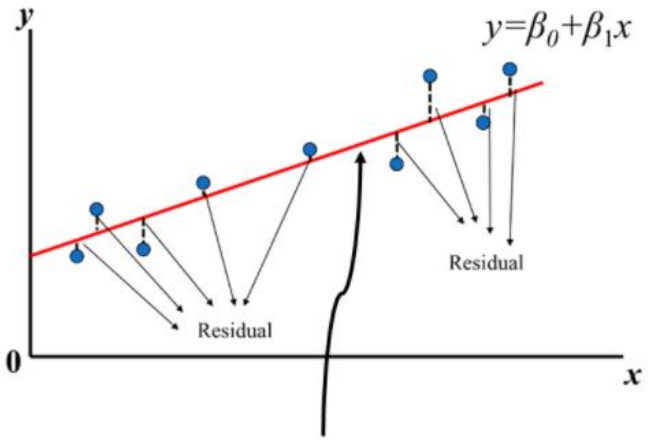
基本概念

- 有多類別的 Logistic Regression，但大部分都是用來解決二分類

$$\sigma\left(f(x)\right)=\begin{cases}0 & f(x)<0\\ 1 & f(x)\geq 0\end{cases} \qquad \sigma\left(f(x)\right)=\frac{1}{1+e^{-f(x)}}=\frac{e^{f(x)}}{1+e^{f(x)}}$$

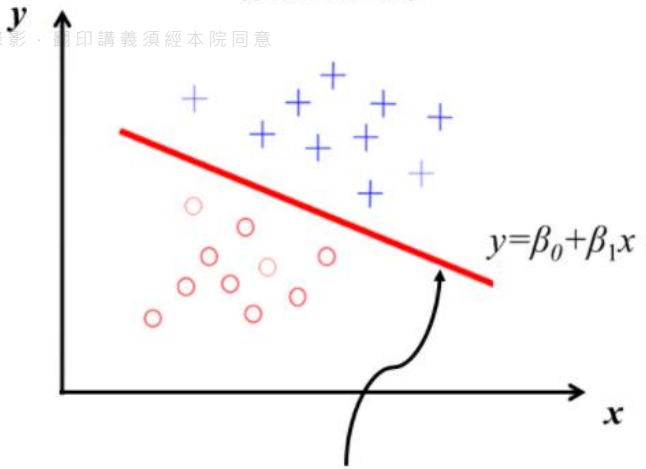
資料來源: Tommy Huang

線性回歸



線性回歸是希望
「找到資料都可以盡量fix的那條紅線」

羅吉斯回歸



羅吉斯回歸希望
「找到那條紅線，讓資料可以區隔開來」

Logistic Regression 優缺點

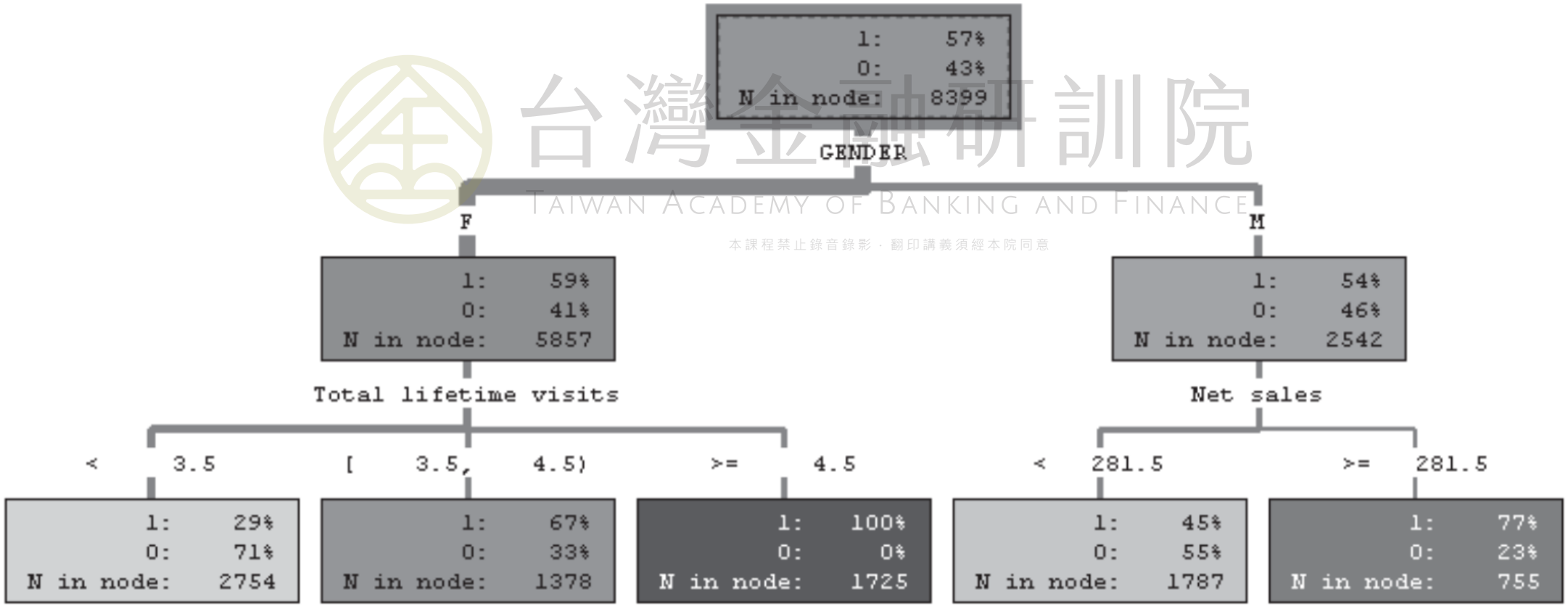
- 優點：
  - 實務上Logistic Regression執行速度非常快
  - 可解釋性高
  - 符合監管要求
- 缺點：
  - 不夠準確

決策樹
(Decision Tree)

DT 算法的目的

- 資訊增益 (越大越好)：原本資訊量 – 分割後資訊量 (越小)
- 衡量指標：熵 (Entropy) 以及 Gini不純度 (Gini Impurity)

並非都為二分割



DT 優缺點

- 優點
  - 模型直觀，便於理解，應用廣泛
  - 算法簡單，容易實現
  - 訓練和預測時，效率較高
  - 不須擔心樣本分布假設與數據問題
- 缺點
  - 缺少足夠的理論支持
  - 如何選擇合適的樹結構對初學者來說比較困惑
  - 決策樹代表性的演算法比較少

DT 優缺點

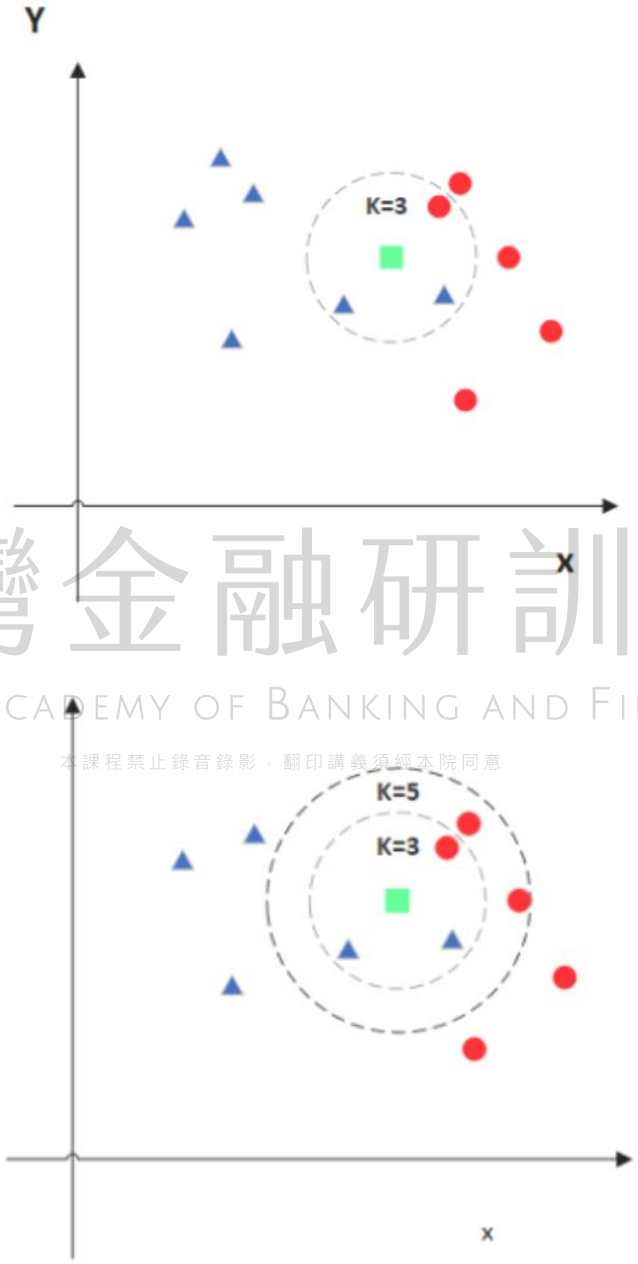
- 優點
  - 模型直觀，便於理解，應用廣泛
  - 算法簡單，容易實現
  - 訓練和預測時，效率較高
  - 不須擔心樣本分布假設與數據問題
- 缺點
  - 缺少足夠的理論支持
  - 如何選擇合適的樹結構對初學者來說比較困惑
  - 決策樹代表性的演算法比較少

K-NN 近鄰分類法 (K-Nearest Neighbor)

Subtitle

算法基本想法

- 根據不同特徵值之間的距離來進行分類的一種簡單的機器學習方法。
- KNN演算法主要應用領域是對未知事物進行分類，即判斷未知事物屬於哪一類，判斷思想是，基於歐幾里得距離，判斷未知事物的特徵和哪一類已知事物的的特徵最接近。



圖片出處

K-NN 優缺點

- 優點
  - 簡單，易於理解，易於實現，無需估計參數，無需訓練
  - 適合對稀有事件進行分類 (尤其當該稀有事件數量少而且種類多)
  - 特別適合於多分類問題 (multi-modal,物件具有多個類別標籤)
- 缺點
  - 實際落地時計算量大，硬體負擔高
  - 可解釋性較差，無法給出決策樹那樣的規則。
- K-NN 的概念廣泛應用在目前很多新算法中，但是單獨使用準確率較低

隨機森林 (Random Forest)

Subtitle

Model Aggregation 的觀念

- 股票推薦
  - Validation: 選擇一個最受信任，對股票預測能力最強的人
  - Uniformly: 投票，一人一票，最終決定出對該支股票的預測。
  - Non-Uniformly (1): 進行投票，只是每個人的投票權重不同。
  - Non-Uniformly (2): 跟 (1) 類似，但是權重不是固定的，根據不同的條件，給予不同的權重。比如如果是傳統行業的股票，那麼給這方面比較厲害的朋友較高的投票權重，如果是服務行業，那麼就給這方面比較厲害的朋友較高的投票權重。
- Aggregation 可以在同一算法不同抽樣，或是在同一樣本不同算法下操作

為何共識決更好

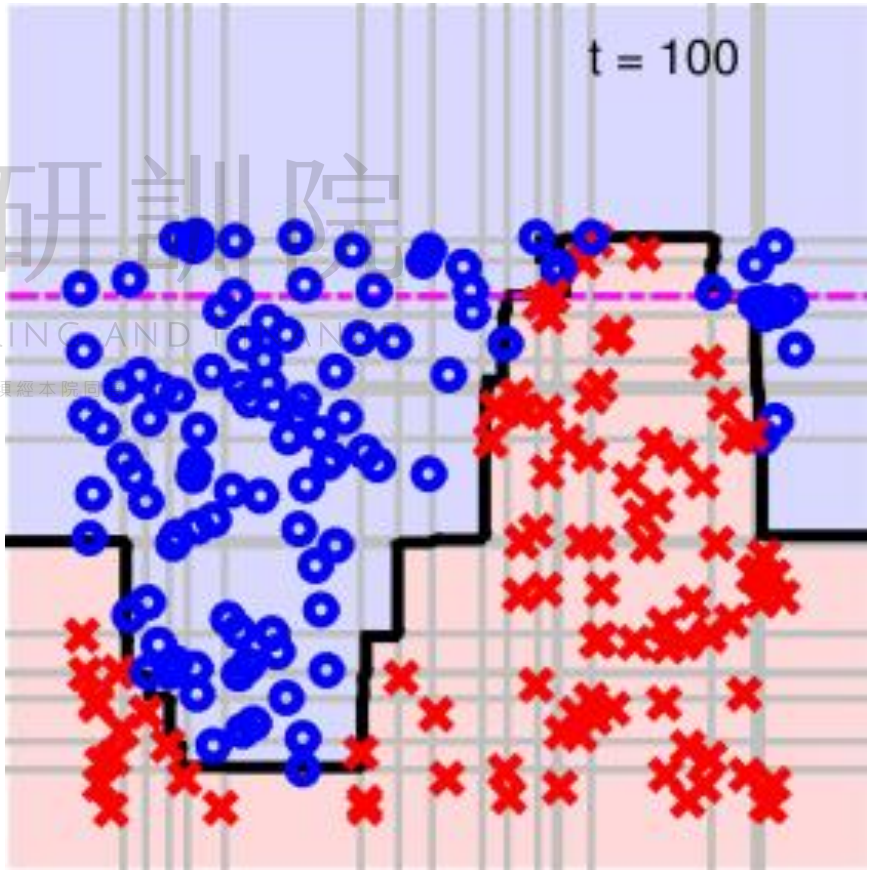
- 模型做出時免不了有 variance + bias，共識決在程度上減少了模型建置的 variance，使預測結果更穩定，證明跳過。

Bagging

- 如何產生不同的 Hypotheses 跟結果 => 從數據著手 (抽樣)
- Bagging (Bootstrap aggregation): 例如原始樣本有10000個，抽取-放回 3000 次，然後產生多個模型後進行 aggregation。

Boosting (李弘毅老師的範例)

- 蘋果分類模型 (有次序的發現重要規則) :
  - (S1) 圓的
  - (S2) 紅色
  - (S3) 也可能是綠色
  - (S4) 上面有梗。
- 透過簡單的 hypotheses ，將所有融合，得到很好的預測模型G。例如，二維平面上簡單的hypotheses (水平線和垂直線) ，這些簡單最終組成的較複雜的分類線能夠較好地將正負樣本完全分開，即得到了好的預測模型
- 關鍵在於每次反覆運算時，放大錯誤樣本，縮小正確樣本，得到不同弱規則，然後合併得到較佳的分類結果。



Random Forest

- Decision Tree + Bagging
- RF 透過驗證機制來避免 Overfitting，並決定特徵重要性
- 隨機抽取一部分特徵。例如，原來有100個特徵，現在只從中隨機選取30個來構成決策樹，那麼每一輪得到的樹都由不同的30個特徵構成，每棵樹都不一樣。假設原來樣本維度是d，則只選擇其中的d' (d' 小於d) 個維度來建立決策樹結構。這類似是一種從d維到d' 維的特徵轉換，相當於是從高維到低維的投影，也就是說d'維z空間其實就是d維x空間的一個隨機子空間 (subspace)。通常情況下，d' 遠小於 d，從而保證演算法更有效率。Random Forest演算法的作者建議在構建 C&RT 每個分支b(x)的時候，都可以重新選擇子特徵來訓練，從而得到更具有多樣性的決策樹。
- 特徵選擇：
  - 提高效率，特徵越少，模型越簡單
  - 正則化，防止特徵過多出現過擬合
  - 去除無關特徵，保留相關性大的特徵，解釋性強
- 實驗發現樹越多，模型越穩定
- 透過隨機種子來增加樹的多樣性，但是隨機種子也會影響分析結果

Q &A

支持向量機 (Support Vector Machine)

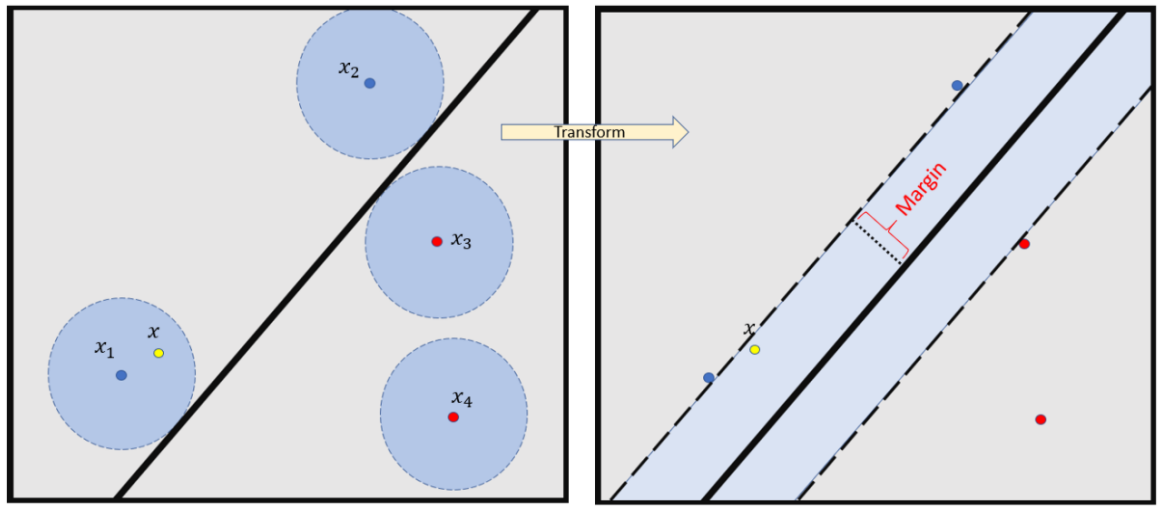
Subtitle

支持向量機的優點

- 潛層學習
- SVM 雖然可以進行多類別與數值預測，但是最強的部份是進行 binary classification
- 在樣本空間建立一個分隔的 hyperplane (解)，完成最大分類。
- 由於解在樣本空間中有無限多種，哪種解是最好的呢 => SVM 算法所優化的問題。

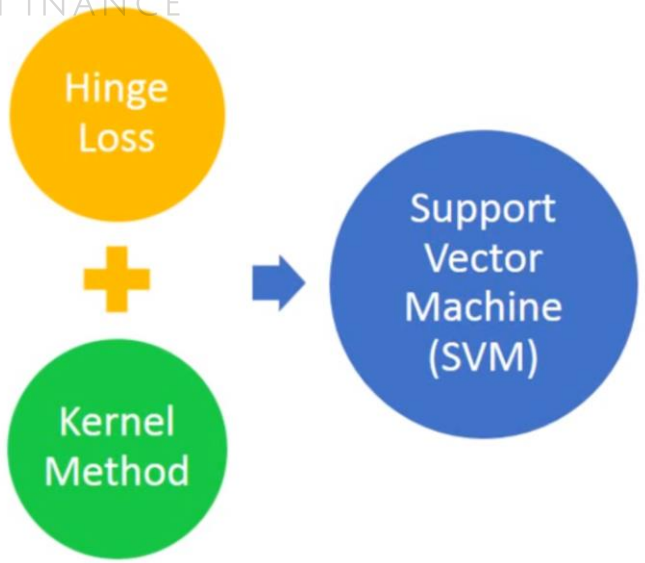
支持向量機的優點

- 在預測中，可能有隨機分布的 Noise 在資料中，這樣的 Noise 不應該影響到我們的分類，則我們會希望最好的 Hyperplane 應該是能夠與各分類資料距離越大越好，也就是說，能夠有更多的誤差容忍度 (i.e. Maximum-Margin Hyperplane) => 最胖的 Hyperplane



支持向量機的優點

- Loss Function: Hinge Loss
- Hyerplane 的建立可以是線性或非線性，由 Kernel 算法決定。
  - Linear
  - Polynomial
  - RBF (Radial-Basis-Function)，高斯核心
  - Sigmoid



核心比較

- Linear
  - 優 – 簡單快速
  - 優 – 具解釋性
  - 缺 – 資料本身必須要是 Linear Separable
- Polynomial
  - 優 – 資料並不一定要 Linear Separable
  - 優 – 多項式轉換的取值可以從資料的特性來判斷
  - 缺 - 太多參數必須做選擇
- RBF (Radial-Basis-Function)，高斯核心
  - 優: Powerful Kernel
  - 缺: 解釋性低
  - 缺: 計算速度慢
  - 缺: Overfitting 的機會很大
- Sigmoid
  - 優: 跟一層 NN相似
  - 缺: 跟 RBF 相似

總結

- 優: 因為 Hinge Loss ，所有的權重都可由少數幾個資料來表示，也因為如此，SVM 模型有很不錯的抗 Outlier 能力。
- 優: 而 Kernel Trick 成功的避開了「該怎麼選擇特徵轉換」的困境，並且成功的大幅降低演算複雜度。
- 優: 綜合以上， SVM 可以用少數資料構造出一個穩定且複雜度不高的模型，在神經網路盛行前，是非常強大的演算法
- 缺點: 如何選擇核心與核心相關參數?

淺層學習 VS 深度學習

- 淺層學習 (簡單的神經網絡與其他相似機器學習算法，SVM 為代表)
- 除文字分析外，實務上發現在金融業數據中淺層學習效果不會比深度學習差
  - 深度學習的優點: 不太需要選擇特徵
  - 金融業數據不大，不須太深的模型

模型實作

使用數據介紹

數據出處
變量描述

- 信貸進件模型
- German Credit
- 20 個輸入變量 (7個連續/13類別)； 1個目標變量
- 數據跟實際銀行數據差別很大，僅能作為簡單教學使用

需要軟體與套件

- 請見課前安裝說明
- Anaconda (python, seaborn, scikit-learn, numpy, pandas, & matplotlib
- Pandas-profiling (Data exploration)
- Imblearn (處理不平衡數據套件)
- Tensorflow & Keras (Deep learning)

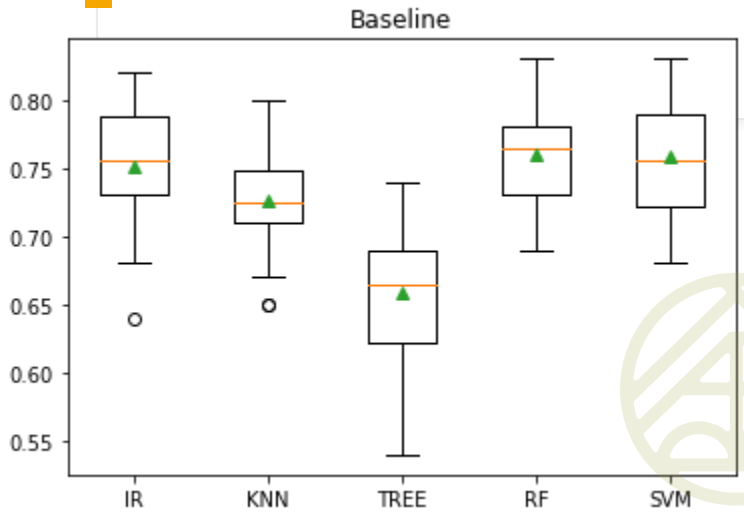
Data visualization

- 好用套件 (pandas-profiling)

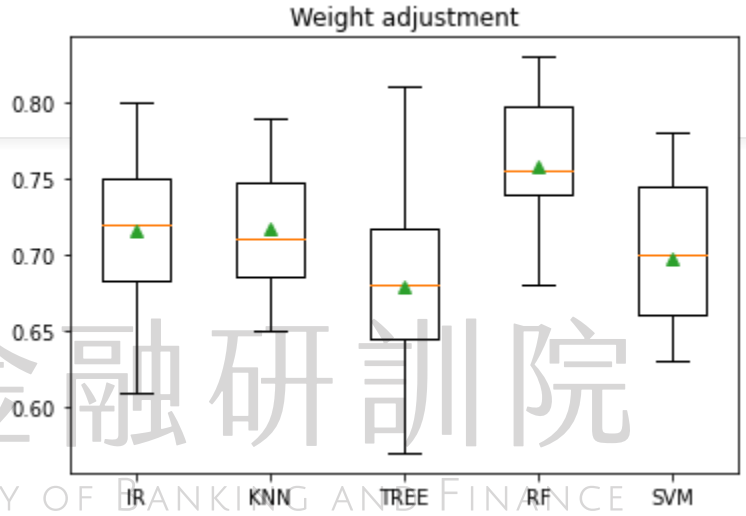
數據處理

- No missing data (無遺失值問題)
- Creditability (0,1) 對調
- Assign variables to correct data types (變量類型)
- One-hot encoding
- Numerical variables => range transformation

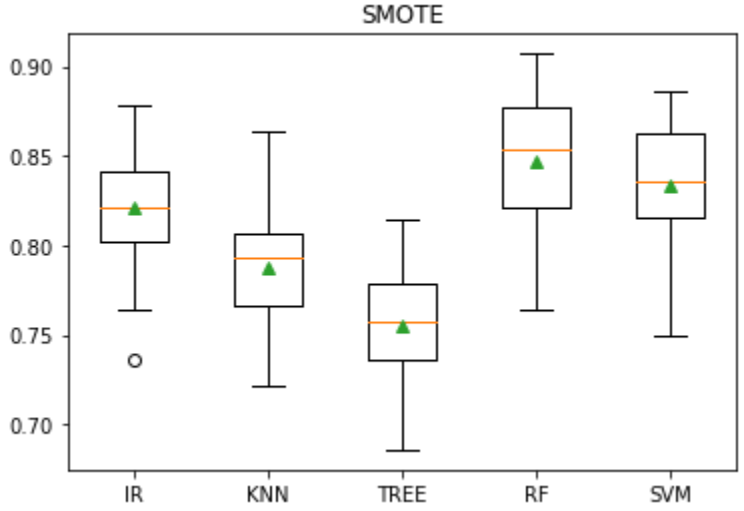
Comparisons



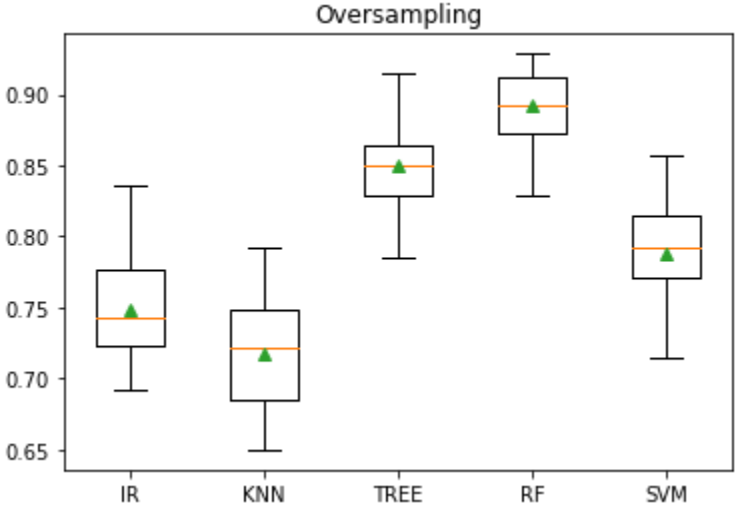
IR 0.751 (0.043)
KNN 0.726 (0.038)
TREE 0.659 (0.047)
RF 0.760 (0.035)
SVM 0.759 (0.042)



IR 0.716 (0.048)
KNN 0.717 (0.040)
TREE 0.679 (0.053)
RF 0.758 (0.039)
SVM 0.698 (0.048)

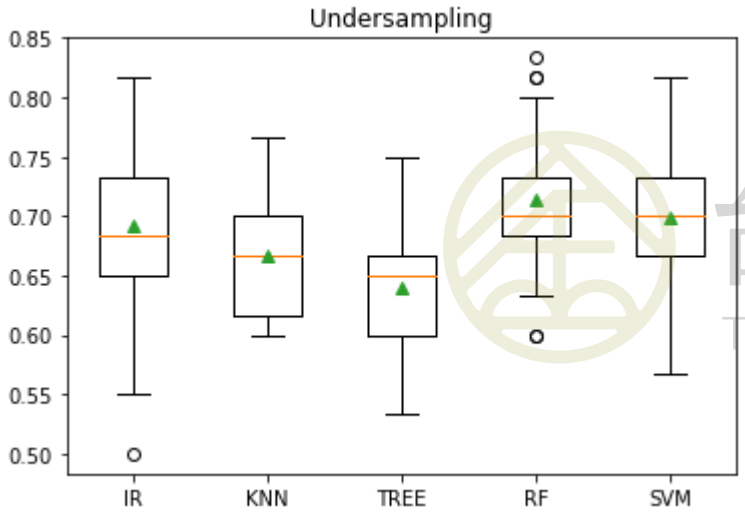


IR 0.822 (0.033)
KNN 0.788 (0.033)
TREE 0.755 (0.032)
RF 0.847 (0.035)
SVM 0.834 (0.033)



IR 0.748 (0.034)
KNN 0.717 (0.038)
TREE 0.850 (0.031)
RF 0.892 (0.024)
SVM 0.789 (0.034)

Comparison_continued



IR 0.693 (0.067)
KNN 0.667 (0.049)
TREE 0.639 (0.052)
RF 0.713 (0.059)
SVM 0.698 (0.060)

傳統建模方法結論

- 以此例子來看，Oversampling & SMOTE 結果較佳
- 實務中 SMOTE 在高維表現會變差
- Weight adjustment 通常較差
- 此次比較並無進行 Hyperparameter adjustments (不在此課程議題)

Q &A

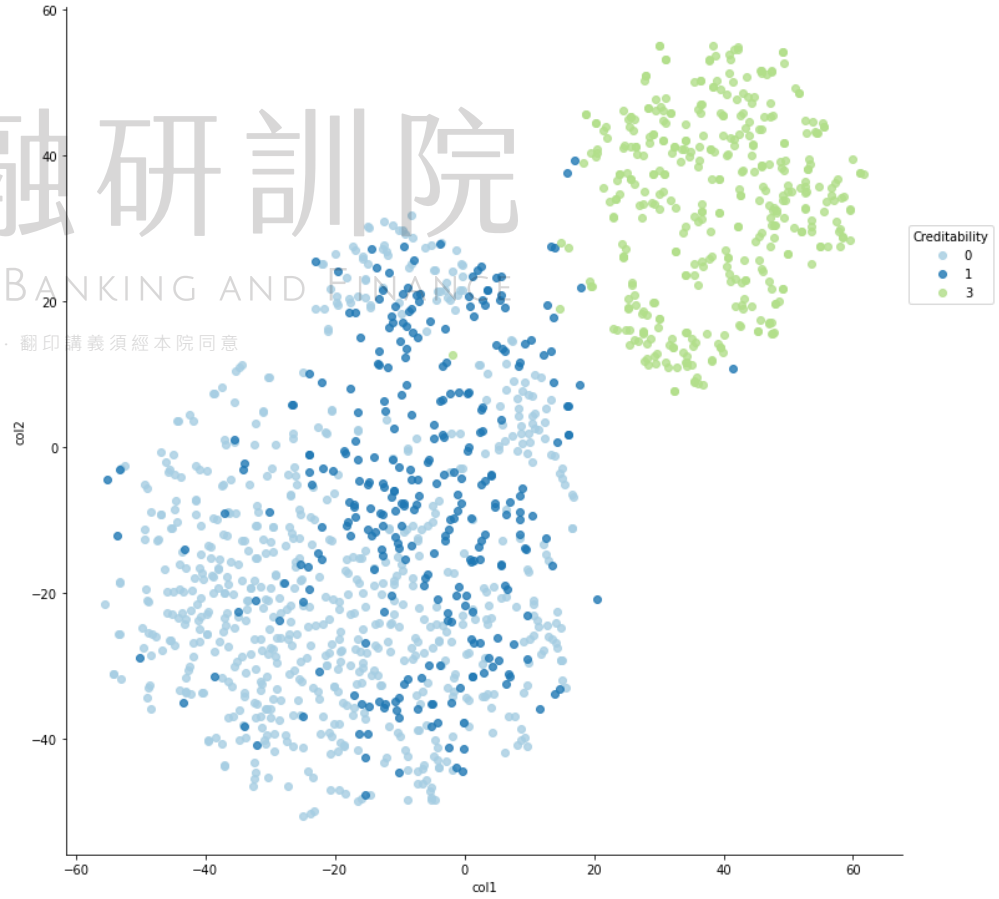
深度學習作法

結果視覺化說明 (t-SNE)

- t-SNE 是一種非線性的機器學習降維方法，主要的功能是把高維的數據降維後投射至 2D 或 3D 的空間進行觀察。

利用 t-SNE 觀察 SMOTE

- 數據處理，將假數據重新編碼



Autoencoder 簡介

- Autoencoder 介紹

Autoencoder 的廣泛應用

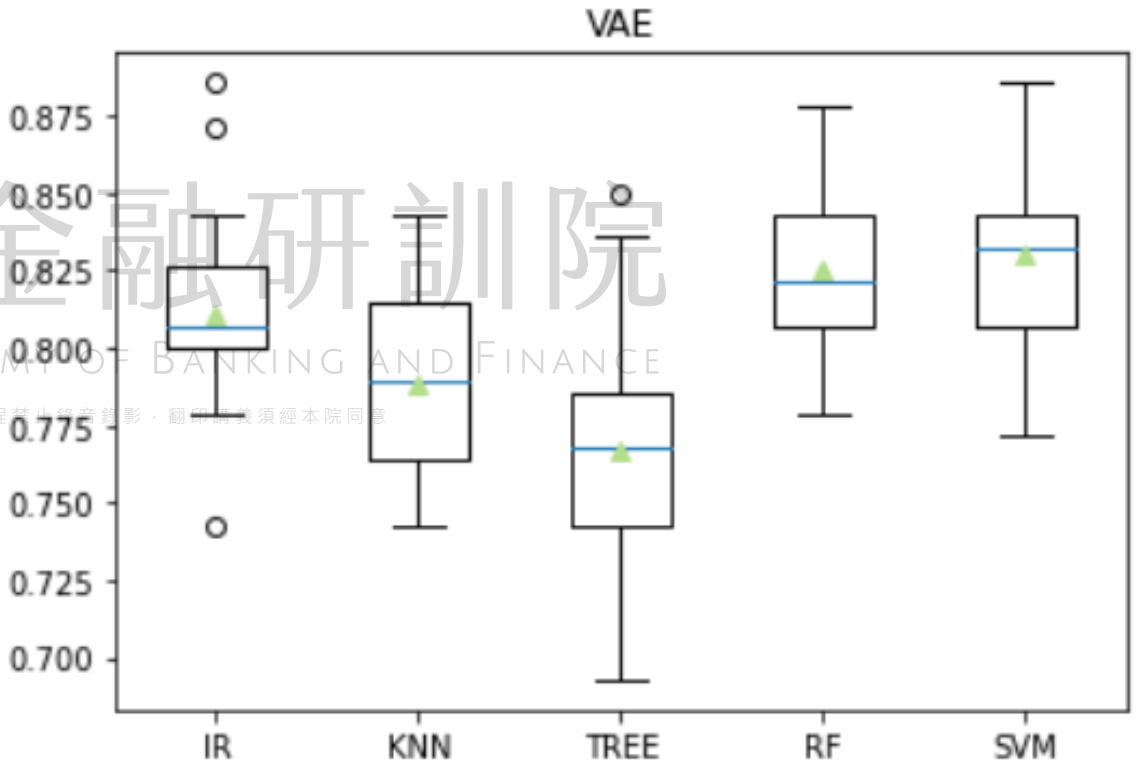
- 模型降維與提升準確度
- 去除數據雜訊，數據、圖片、文字 (如文字分析中 Bert)
  - Token Embedding：就是對輸入的每次單詞進行Embedding。
  - Segment Embedding：標記輸入的Token是屬於句子A還是句子B。
  - Position Embedding：具體標記每一個Token的位置。
- 生成假樣本 (Variational AE)
- 稀疏表示 (Sparse AE)，選擇更具代表性的特徵 (例如 LSTM 當中的 attention 機制)。

模型實作

Variational Autoencoder

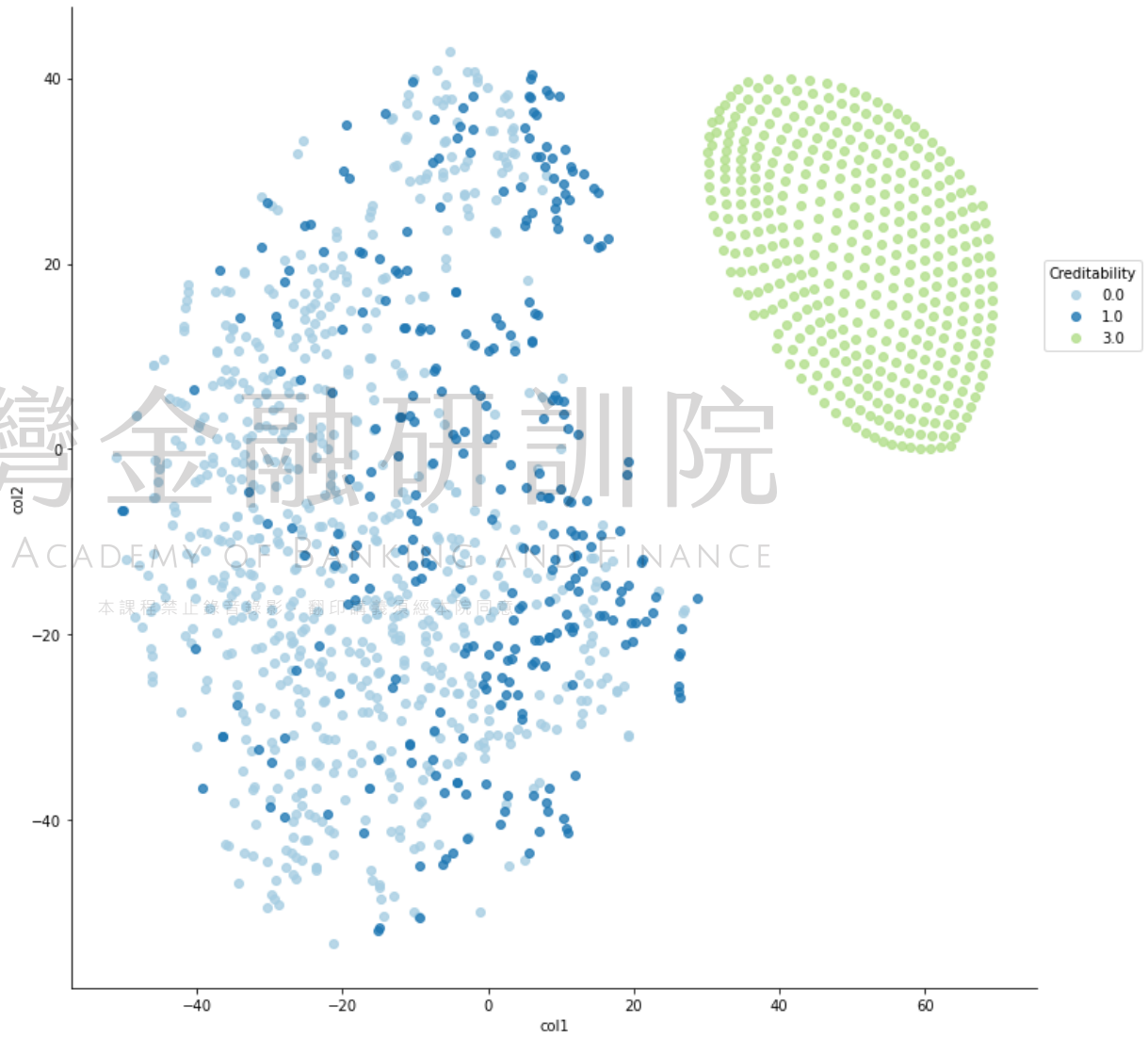
- 利用 VAE 產生假數據，優化模型結果
- VAE 整體準確度略差於 SMOTE，但標準差較小
- 訓練結果顯示 VAE 有學習到重點，並改善模型結果

IR 0.811 (0.027)
KNN 0.789 (0.028)
TREE 0.767 (0.033)
RF 0.826 (0.024)
SVM 0.830 (0.028)



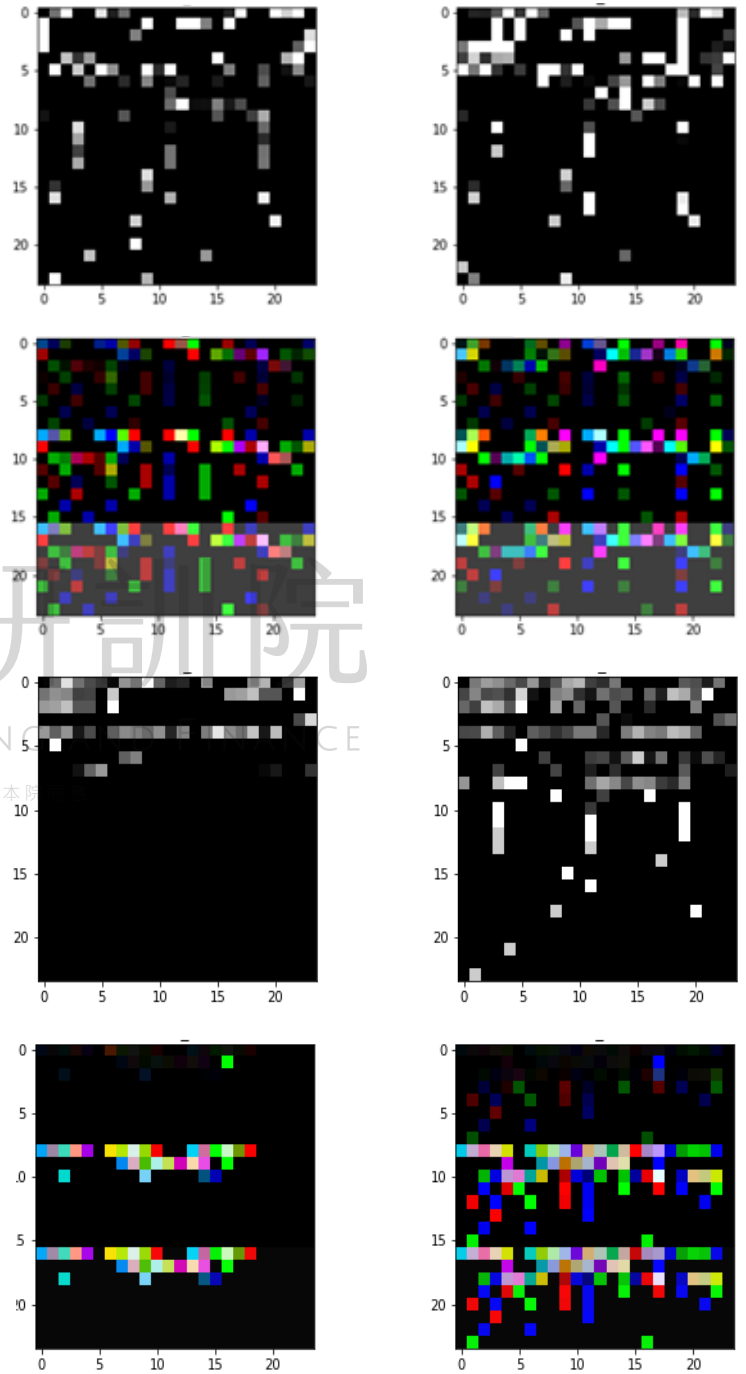
利用 t-SNE 觀察 VAE

- 數據處理，將假數據重新編碼
- 遠離正常樣本，以此改善模型結果。
- 因為數據集有包含較高比例類別量變量，從 t-SNE 分布上與原始數據差別較大。
- 根據實務經驗，VAE 在高維數據表現較佳



其他解決方案

- GAN (對抗網路)
- CNN, GCN or self-constructed models (其他技術手段)
  - 模型方法是通用的，CNN 並非只能處理圖形，LSTM 也不是只能用在時間序列



最新的技術自動化進展

- AutoML
  - H20, TPOT, Auto-sklean (模型解釋完全不重要)
- Auto Hyperparameter Tuning (計算成本高)
  - Ray Tune, Optuna, HyperOpt, Scikit-Optimize , Microsoft's NNI , AI Platform Vizier ,
  - AWS Sage Maker , Azure Machine Learning Studio
- Auto Feature Engineering (計算成本高) (Tools)
  - Feature Selection: Xverse
  - Feature Extraction: Automated feature engineering tool (著重在特徵抽取)
- 整體流程 (https://github.com/alteryx/compose)

Q&A

andy.ttu@gmail.com