機器學習基礎與演算法

Chapter 8　非監督式學習 (Unsupervised Learning)

Chapter 9　維度縮減 (Dimension Reduction)

Chapter 10　Kaggle 實戰簡介

講師投影片
課程投影片
資料與程式碼
今日Playlist

「版權聲明頁」

課程內容

8.非監督式學習 (Unsupervised Learning)

- 8-1 [理論講授] 非監督學習式學習

9. 維度縮減 (Dimension Reduction)

- 9-1 [理論講授] 維度縮減
- 9-2 [實作講授] 主成份分析 (PCA)

10. Kaggle 實戰

- 10-1 [理論講授] Kaggle 平台介紹
- 10-2 [實作課程] Kaggle 機器學習實戰

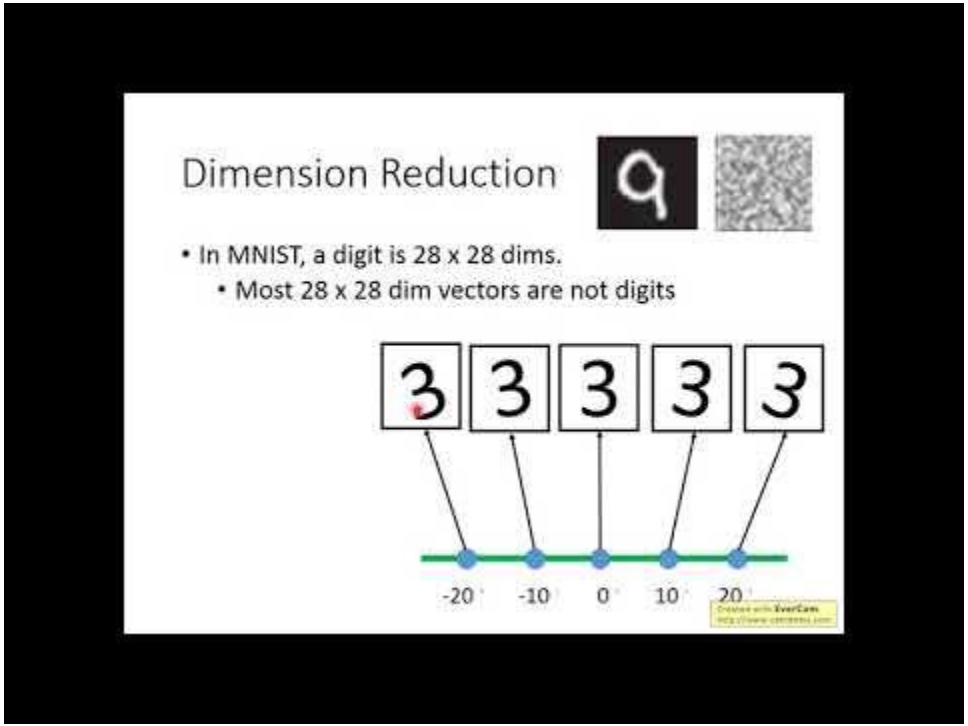
Chapter 8　非監督式學習 (Unsupervised learning)

- 範例程式(example)的檔名會以**藍色字體**顯示且旁邊附上
- 練習(exercise)的檔案以**紅色字體**顯示且旁邊附上

Section 8-1　[理論講授] 非監督學習式學習

Chapter 9　維度縮減 (Dimension Reduction)

Section 9-1　[理論講授] 維度縮減



Section 9-2　[實作講授] 主成份分析 (PCA)

- 實務上我們經常遇到資料有非常多的 features，有些 features 可能高度相關，有什麼方法能夠把高度相關的 features 去除？
- PCA 透過計算 eigen-value, eigen-vector，可以將原本的 features 降維至特定的維度
  - 原本 Data 有 100 個 features，透過 PCA，可以將這 100 個 features 降成 2 個 feautres
  - 新 features 為舊 features 的線性組合

新 feaures 彼此不相關

The original variables is noted as $x_1, x_2, \ldots, x_n$, and the new variables can be represented as

$$
\begin{aligned}
z_1 &= a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\
z_2 &= a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\
&\vdots \\
z_n &= a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n
\end{aligned}
\quad \Big\} \text{ Uncorrelated}
$$

PCA in Scikit-learn

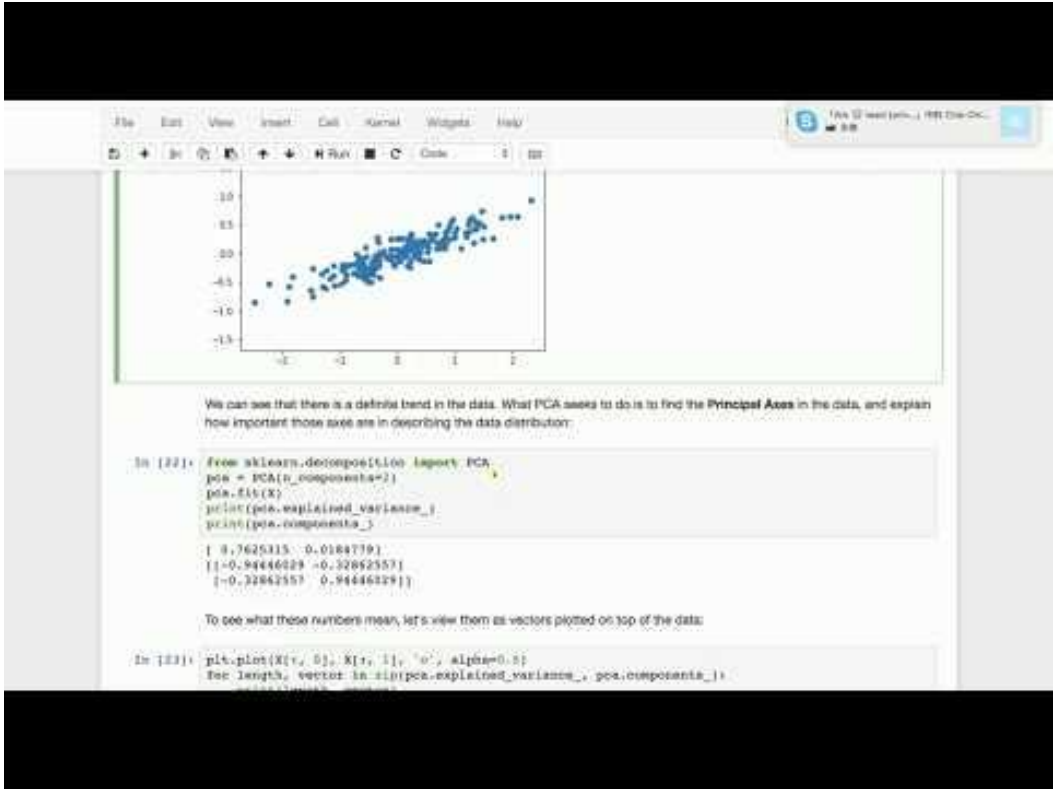
```
from sklearn.decomposition import PCA

pca = PCA(n_componets=2)

X_reduct = pca.fit_transform(X) #X.shape=(200, 64)

print(X_reduct.shape)  #(200, 2)
```

PCA 實戰



練習

- **使用 digits dataset** 

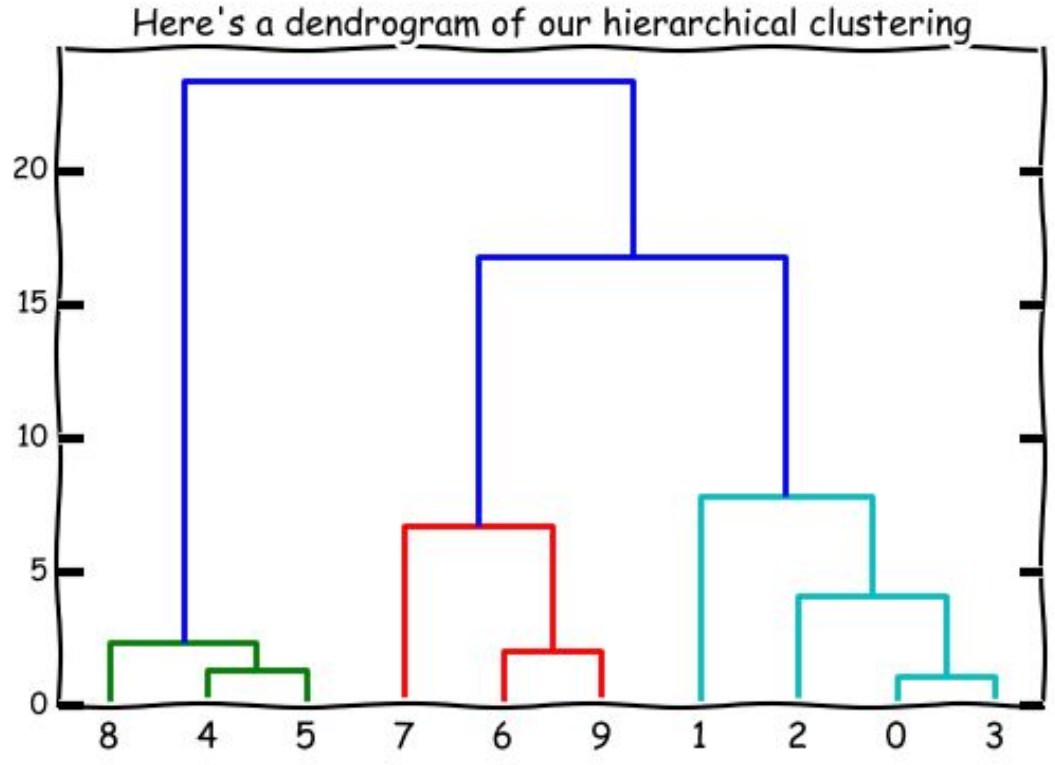
，比較如果將資料降維之後再訓練模型，準確度是否會提升

階層式分析

- 不需指定分群的數量
  1. 每筆資料視為獨立一群
  2. 計算每兩群之間的**距離**
  3. 將最近的兩群合併成一群
  4. 重複 2,3 直到所有資料合併為同一群為止
- 計算距離的方式有
  - `'complete'`: cluster 中，最遠兩點的距離
  - `'single'`: cluster 中，最近兩點的距離
  - `'average'`: cluster 中，所有點的距離平均

階層分析後的樹狀圖 (dendrogram)

- 可定義 4, 5 是一群，或 8, 4 ,5 是一群，端看距離怎麼衡量



練習

- 請參考 session3 中的 **hierarchical_clustering_example**，試著理解 code

補充閱讀

- PCA
- Hierarchical

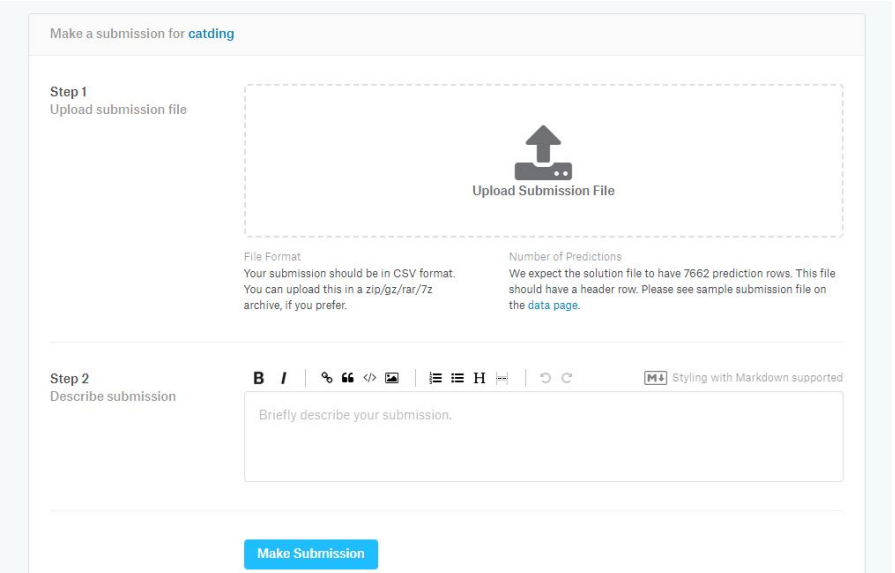
Chapter 10　Kaggle 實戰簡介

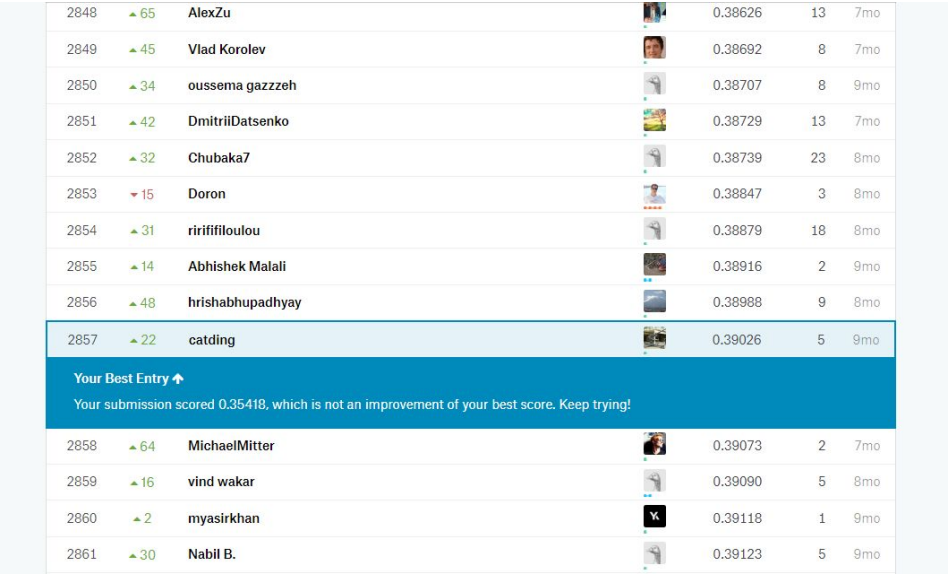
Section 10-1　[理論講授] Kaggle 平台介紹

Kaggle 平台簡介

- Kaggle 為一個全球性的資料科學競賽網站，任何人都可以上傳數據資料來舉辦比賽，很多公司會發布一些接近真實業務的問題，並提供獎金來吸引愛好數據科學的人來一起解決問題
- 有些公司甚至會用 Kaggle 上的排名來評估應徵者

上傳預測結果就會看到排名





2848	▲65	AlexZu	0.38626	13	7mo
2849	▲45	Vlad Korolev	0.38692	8	7mo
2850	▲34	oussema gazzzeh	0.38707	8	9mo
2851	▲42	DmitriiDatsenko	0.38729	13	7mo
2852	▲32	Chubaka7	0.38739	23	8mo
2853	▼15	Doron	0.38847	3	8mo
2854	▲31	rirififiloulou	0.38879	18	8mo
2855	▲14	Abhishek Malali	0.38916	2	9mo
2856	▲48	hrishabhupadhyay	0.38988	9	8mo
2857	▲22	catding	0.39026	5	9mo

Your Best Entry ↑

Your submission scored 0.35418, which is not an improvement of your best score. Keep trying!

2858	▲64	MichaelMitter	0.39073	2	7mo
2859	▲16	vind wakar	0.39090	5	8mo
2860	▲2	myasirkhan	0.39118	1	9mo
2861	▲30	Nabil B.	0.39123	5	9mo

機器學習 Kaggle 實戰：波士頓房價預測

題目連結：House Prices: Advanced Regression Techniques

- 總共有 76 個 features, features 的說明 **請點此**
- 每天最多上傳 5 次結果
- 請各位試著完成本題練習, 請注意期中考可能包含類似 (機器學習類) 題型
- 歡迎各位寫作時討論, 或參考本課程說明以及參閱 **競賽Kernel**, 建議以能讀懂並能自己修改 Code 為目標

(optional) 完成 scikit-learn-practice 比賽

題目連結：scikit-learn-practice

- 這是單純讓大家熟悉 Scikit-learn 的比賽。總共有一千筆訓練資料、40個 features，簡單的二元分類問題。可在房產預測競賽後作為練習，以提升自己實力
- 資料並沒有提供 features 的意義，純粹是讓大家練習 features scaling、建模、調參數等步驟
- 每天最多上傳 10 次結果
- 請在 private / public leaderboard 上取得 0.8 以上的準確度。達標代表你對 Scikit-learn 的操作有一定水準囉！

資料下載

可至競賽中 **Data** 頁面下載, 或直接使用課程練習題目錄資料

	..	seconds ago
☐	data-science-london-scikit-learn	2 hours ago
☐	house_prices_advanced_regression_techniques	2 minutes ago

	..
☐	test.csv
☐	train.csv
☐	trainLabels.csv

在開始前...

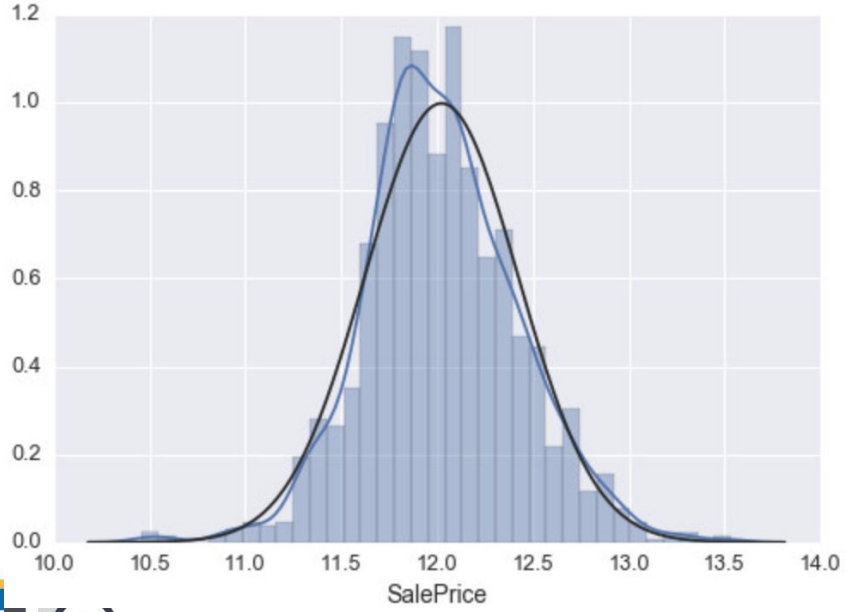
- 以下約略是整個資料分析競賽的流程
  - 資料清洗與轉換
  - 探索式資料分析 (EDA)
  - 特徵工程 (feautre engineering)
  - 建立模型
  - 調整參數
  - 上傳結果
- 以下替各位做簡單的整理

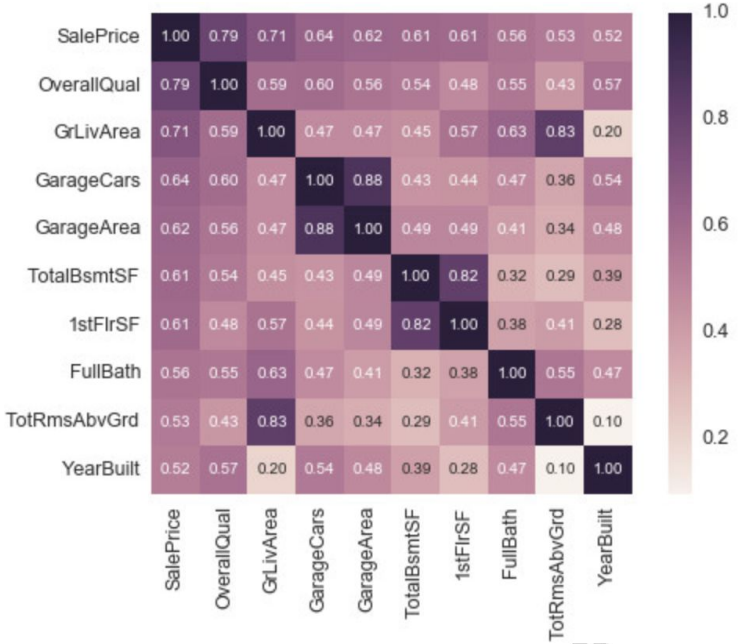
資料清洗與轉換

- 注意是否有遺失值 (missing value)，該用什麼方式填補
- 注意是否有 outliers。實務上依照不同 task，outliers 的定義不同，須自行判斷
- 一些右偏分佈的 feature，可透過取 log 將其轉為常態分佈
- 類別的變數可透過 one-hot encoding 轉為數字
- 類別太多可試著將相近的類別歸為一類
- 如果會用到一些算法像是 PCA, regression，要記得將資料進行標準化 (normalization)

探索式資料分析 (EDA)

- 繪製變數的分佈圖 (hist)、盒鬚圖 (box)、相關係數圖等等





特徵工程 (feature engineering)

- 特徵建立是一門藝術，依靠創意＋經驗＋專家領域知識，沒有標準答案。能找到最關鍵的 feature，即使用普通的模型，也能榜上有名
- 常見的做法：feature 之間的交互關係，例如相乘、相除、取 log、平方、三次方等等

建立模型

- 每個模型都可以嘗試看看，但建議先從簡單的模型開始！
  e.g. linear regression，並把結果當成 baseline 參考
- 後續的模型如果結果有比 baseline 還差，就要注意是否參數有問題，甚至 code 是否寫錯

調整參數

- 強大的模型伴隨著許多參數要調控，但如果一直使用固定的資料來進行調參，就有可能發生 Overfiithg 的情形
- 善用 cross-validation + grid search 來尋找最好的參數，再使用這個參數，重新訓練你的模型，細節請參考正確調參數的方式

常見問題

- 別人的模型比較好？那是因為別人用了『Ensemble』！把 RandomForest + XGBoost + GradientBoosting 的結果全部合併起來，通常會再提昇一些些 performance
- 若是模型在自己切的 testing data 表現很好，但是上傳後 publicboard 的分數卻很低，那很有可能是 Overfitting
- Data 數量較少時，請務必使用 cross-validation 評估結果
- 分類問題如果遇到 data imbalance，可嘗試使用 oversampling 或 undersampling 的方式改善，可參考連結

Machine Learning Algorithms Cheat Sheet