機器學習基礎與演算法

Chapter 1　機器學習概論

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

1. 機器學習概論

1-1 [理論講授] 人工智慧與
迴歸(regression)概論

1-2 [實作課程] 細說基礎流程

- Step 1 定義問題
- Step 2 蒐集, 清理資料
- Step 3 選擇及建立模型
- Step 4 分析結果及修正模型

1-3 [理論講授] 正則化(Regularization)

Code 放在Hub中的course內

- 為維護課程資料，courses中的檔案皆為read-only，如需修改請cp至自身環境中
- 打開terminal，輸入

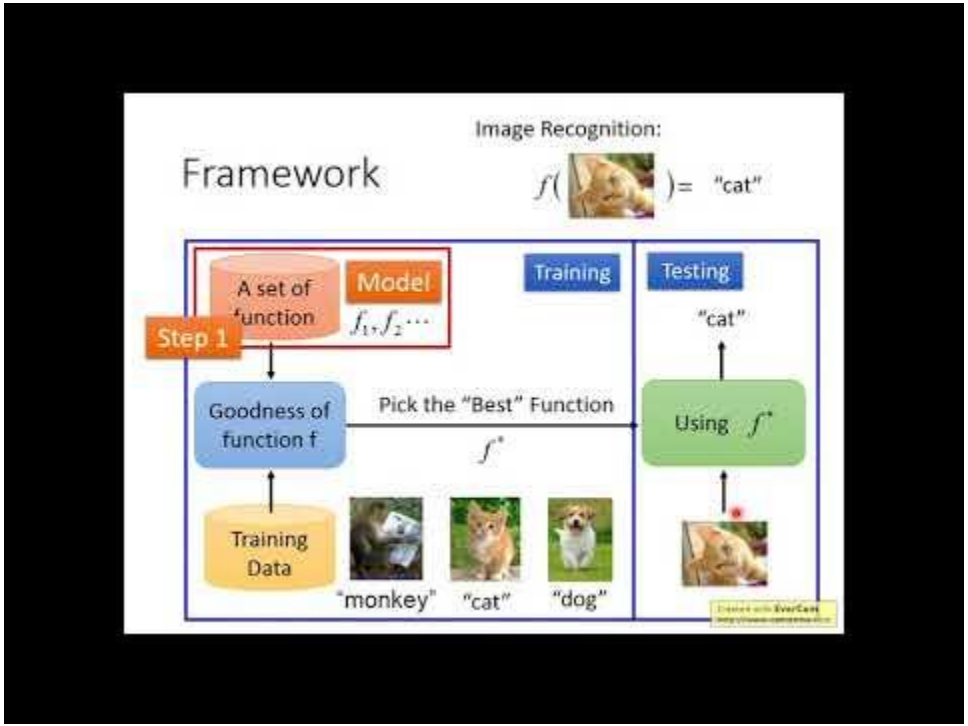
cp -r courses-tpe/ML/Chapter1 <存放至本機的名稱>

Chapter 1　機器學習概論

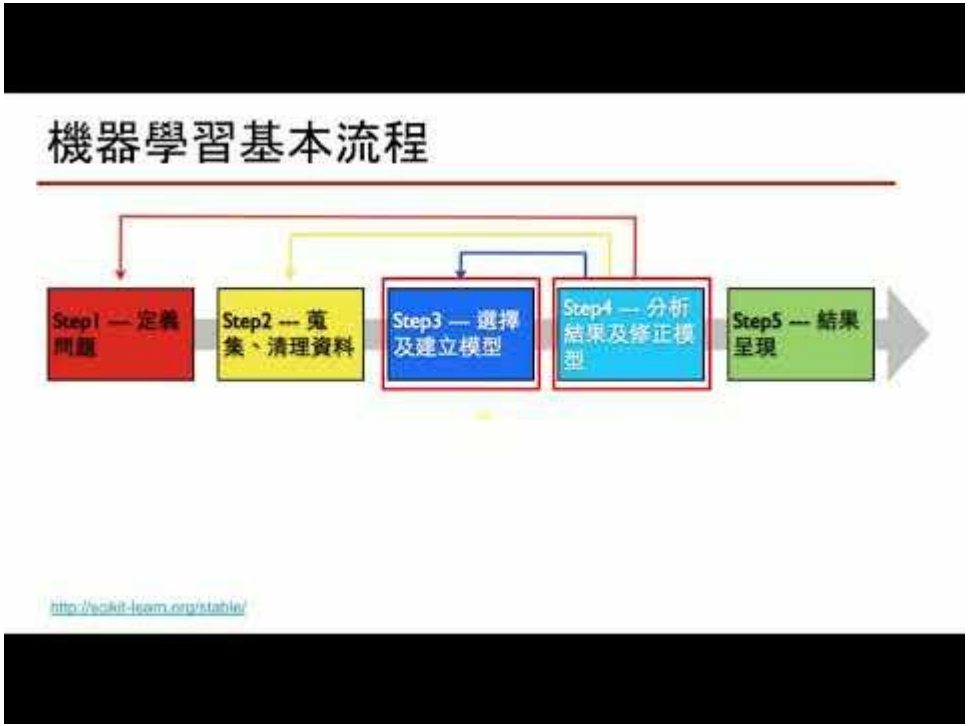
- 範例程式(example)的檔名會以**藍色字體**顯示且旁邊附上 

- 練習(exercise)的檔案以**紅色字體**顯示且旁邊附上 

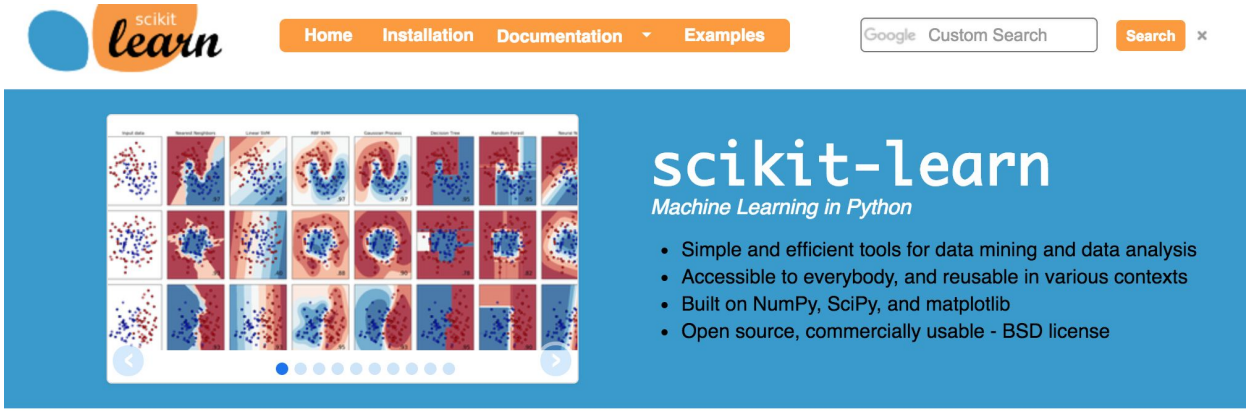
Section 1-1　[理論講授] 人工智慧與迴歸概論



Section 1-2　[實作課程] 細說基礎流程



基礎流程



http://scikit-learn.org/stable/

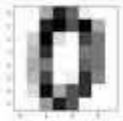
Step 1 - 定義問題：Scikit Learn 提供的數據庫

Scikit Learn 提供的數據庫 Example

- 0-Load dataset from Scikit Learn Example

```
from sklearn import datasets
digits = datasets.load_digits()
```

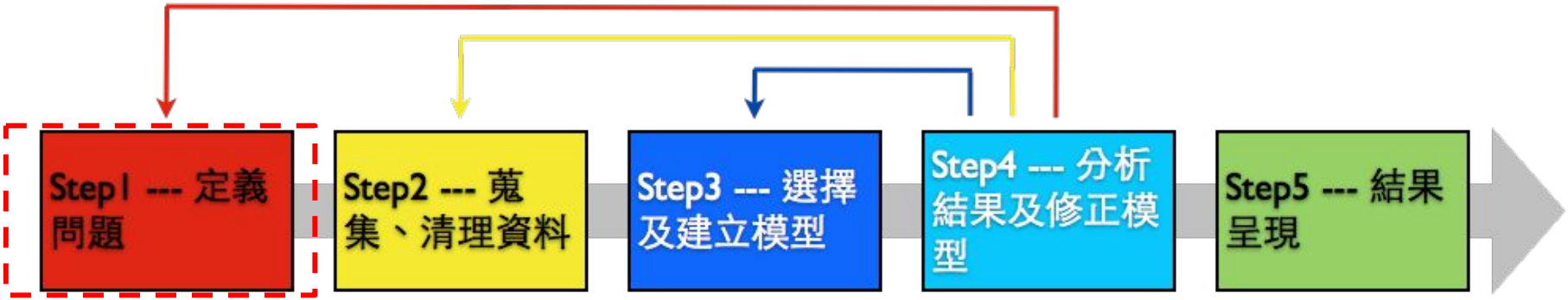
X → y



X_data: digits.data
features數量: 64
sample num: 1797
kind: continuous

y_data: digits.target
label數量: 1
sample num: 1797
kind: categorical (0,1,2,...9)

Scikit Learn 提供的數據庫 (1/3)



sklearn.datasets : Datasets

The sklearn.datasets module includes utilities to load datasets, including methods to load and fetch popular reference datasets. It also features some artificial data generators.

User guide: See the Dataset loading utilities section for further details.

Loaders

datasets.clear_data_home ([data_home])	Delete all the content of the data home cache.
datasets.dump_svmlight_file (X, y, f[, ...])	Dump the dataset in svmlight / libsvm file format.
datasets.fetch_20newsgroups ([data_home, ...])	Load the filenames and data from the 20 newsgroups dataset.
datasets.fetch_20newsgroups_vectorized ([...])	Load the 20 newsgroups dataset and transform it into tf-idf vectors.
datasets.fetch_california_housing ([...])	Loader for the California housing dataset from StatLib.
datasets.fetch_covtype ([data_home, ...])	Load the covertype dataset, downloading it if necessary.
datasets.fetch_kddcup99 ([subset, data_home, ...])	Load and return the kddcup 99 dataset (classification).
datasets.fetch_lfw_pairs ([subset, ...])	Loader for the Labeled Faces in the Wild (LFW) pairs dataset
datasets.fetch_lfw_people ([data_home, ...])	Loader for the Labeled Faces in the Wild (LFW) people dataset
datasets.fetch_mldata (dataname[, ...])	Fetch an mldata.org data set
datasets.fetch_olivetti_faces ([data_home, ...])	Loader for the Olivetti faces data-set from AT&T.
datasets.fetch_rcv1 ([data_home, subset, ...])	Load the RCV1 multilabel dataset, downloading it if necessary.
datasets.fetch_species_distributions ([...])	Loader for species distribution dataset from Phillips et.
datasets.get_data_home ([data_home])	Return the path of the scikit-learn data dir.
datasets.load_boston ([return_X_y])	Load and return the boston house-prices dataset (regression).
datasets.load_breast_cancer ([return_X_y])	Load and return the breast cancer wisconsin dataset (classification).

Example:

```
from sklearn import datasets
digits = datasets.load_digits()
```

http://scikit-learn.org/stable/modules/classes.html#module-sklearn.datasets

Scikit Learn 提供的數據庫 (2/3) - Example

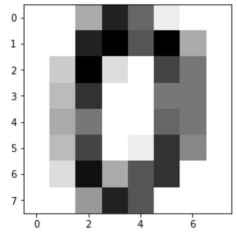
- **0-Load dataset from Scikit Learn Example**

```
from sklearn import datasets
digits = datasets.load_digits()
```

X → y

```
import matplotlib.pyplot as plt
print('answer:%d'%digits.target[0])
plt.imshow(digits.images[0], cmap=plt.cm.binary, interpolation='nearest')
plt.show()
```

```
answer:0
```

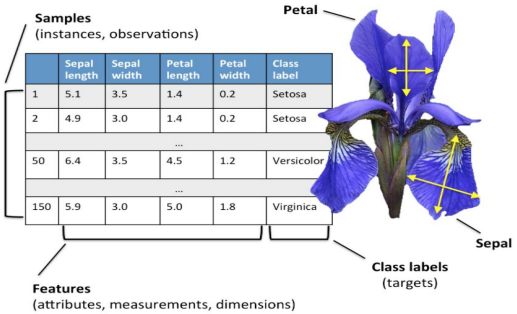


X_data: digits.data
features數量: 64
sample num: 1797
kind: continuous

y_data: digits.target
label數量: 1
sample num: 1797
kind: categorical (0,1,2,...9)

Scikit Learn 提供的數據庫 (3/3) - Exercise

- **0-Load dataset from Scikit Learn Exercise**
  - 透過資料的描述對資料進行了解
  - 了解4個features分別對應什麼名稱
  - 了解三種labels分別對應什麼花名



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

X → y

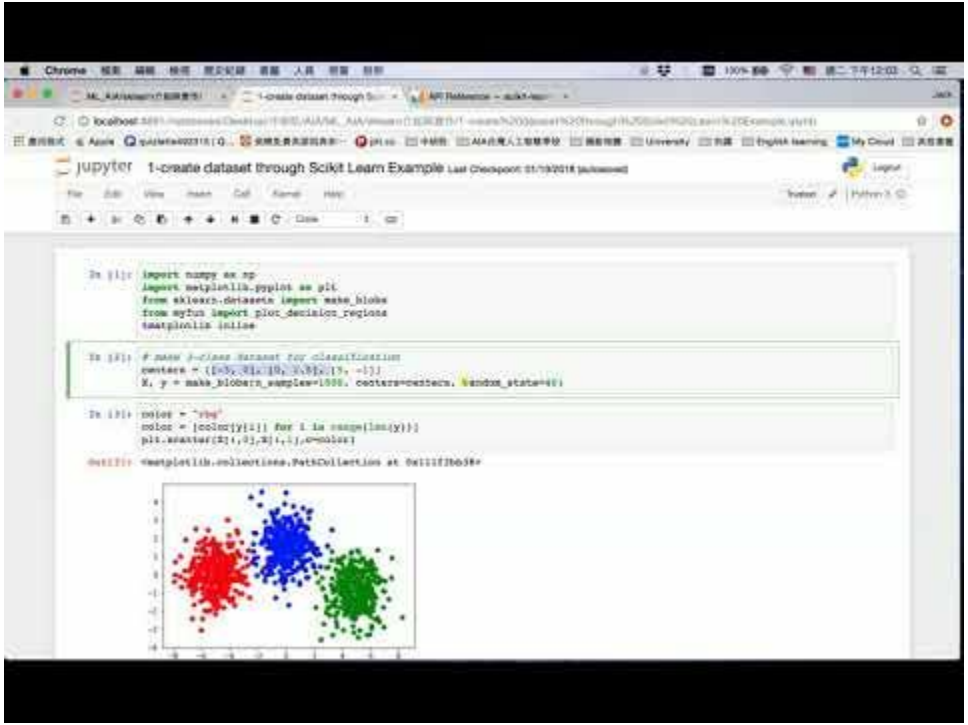
'sepal length (cm)','sepal width (cm)',
'petal length (cm)', 'petal width (cm)'

'setosa', 'versicolor', 'virginica'

X_data: iris.data
features數量: 4
sample num: 150
kind: continuous

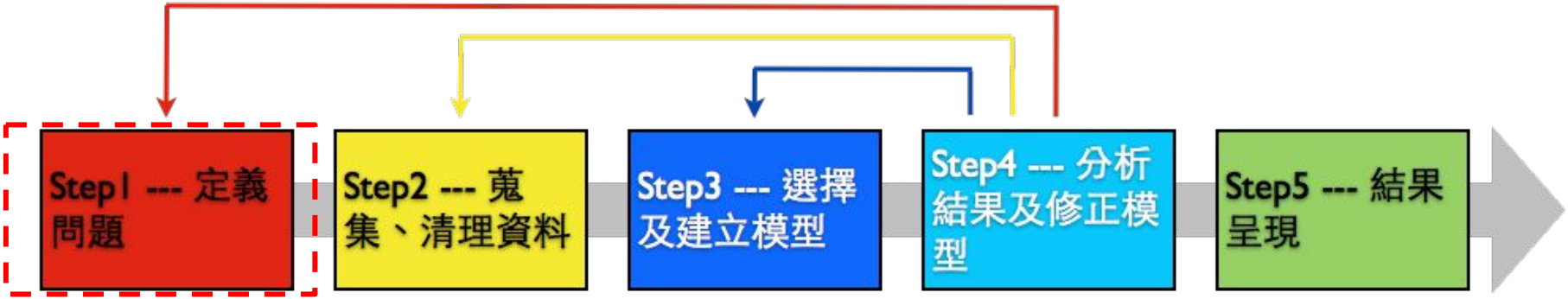
y_data: iris.target
label數量: 1
sample num: 150
kind: categorical (0,1,2)

Step 1 - 定義問題：虛擬樣本的生成



Scikit Learn 提供虛擬資料創建方法

1-Create Dataset through Scikit Learn Example

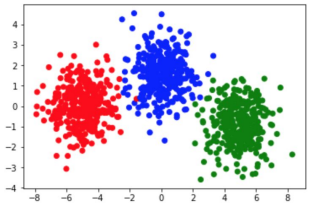


Samples generator

`datasets.make_biclusters` (shape, n_clusters)	Generate an array with constant block diagonal structure for biclustering.
`datasets.make_blobs` ([n_samples, n_features, …])	Generate isotropic Gaussian blobs for clustering.
`datasets.make_checkerboard` (shape, n_clusters)	Generate an array with block checkerboard structure for biclustering.
`datasets.make_circles` ([n_samples, shuffle, …])	Make a large circle containing a smaller circle in 2d.
`datasets.make_classification` ([n_samples, …])	Generate a random n-class classification problem.
`datasets.make_friedman1` ([n_samples, …])	Generate the “Friedman #1” regression problem
`datasets.make_friedman2` ([n_samples, noise, …])	Generate the “Friedman #2” regression problem
`datasets.make_friedman3` ([n_samples, noise, …])	Generate the “Friedman #3” regression problem
`datasets.make_gaussian_quantiles` ([mean, …])	Generate isotropic Gaussian and label samples by quantile
`datasets.make_hastie_10_2` ([n_samples, …])	Generates data for binary classification used in Hastie et al.
`datasets.make_low_rank_matrix` ([n_samples, …])	Generate a mostly low rank matrix with bell-shaped singular values
`datasets.make_moons` ([n_samples, shuffle, …])	Make two interleaving half circles
`datasets.make_multilabel_classification` ([…])	Generate a random multilabel classification problem.
`datasets.make_regression` ([n_samples, …])	Generate a random regression problem.
`datasets.make_s_curve` ([n_samples, noise, …])	Generate an S curve dataset.
`datasets.make_sparse_coded_signal` (n_samples, …)	Generate a signal as a sparse combination of dictionary elements.
`datasets.make_sparse_spd_matrix` ([dim, …])	Generate a sparse symmetric definite positive matrix.
`datasets.make_sparse_uncorrelated` ([…])	Generate a random regression problem with sparse uncorrelated design
`datasets.make_spd_matrix` (n_dim[, random_state])	Generate a random symmetric, positive-definite matrix.
`datasets.make_swiss_roll` ([n_samples, noise, …])	Generate a swiss roll dataset.

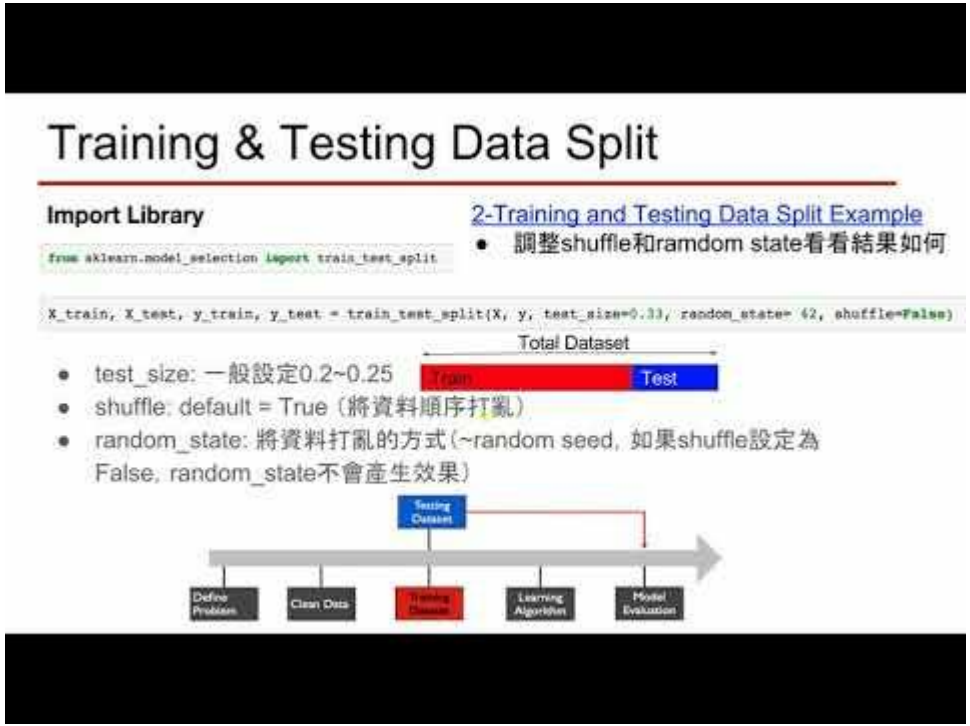
```
# make 3-class dataset for classification
centers = [[-5, 0], [0, 1.5], [5, -1]]
X, y = make_blobs(n_samples=1000, centers=centers, random_state=40)
```

http://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_blobs.html#sklearn.datasets.make_blobs

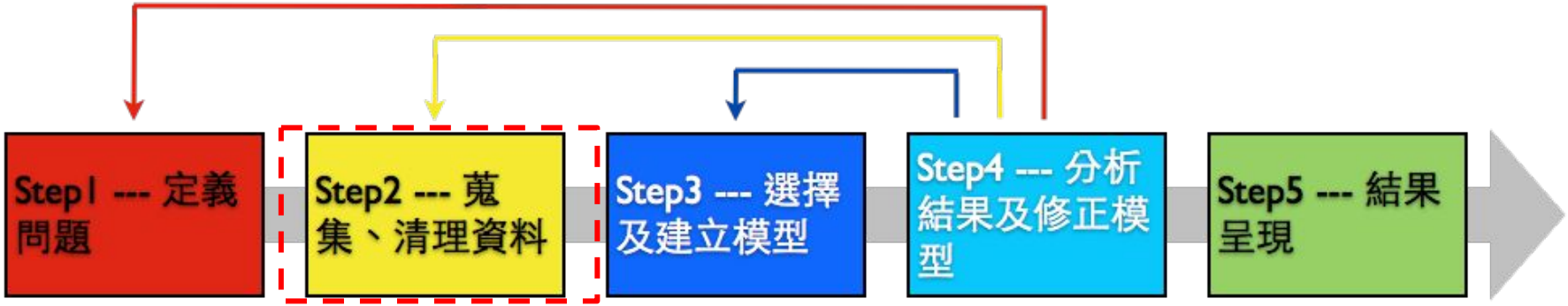


http://scikit-learn.org/stable/modules/classes.html#module-sklearn.datasets

Step 2 - 蒐集, 清理資料



Scikit Learn 提供資料前處理常見方法



- Training & Testing Data Split
- Data Normalization
- One Hot Encoding

Training & Testing Data Split

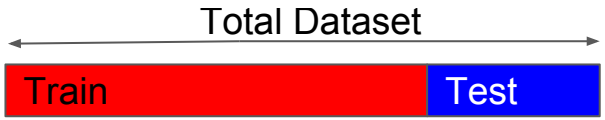
2-Training and Testing Data Split Example

- 調整shuffle和random state看看結果如何

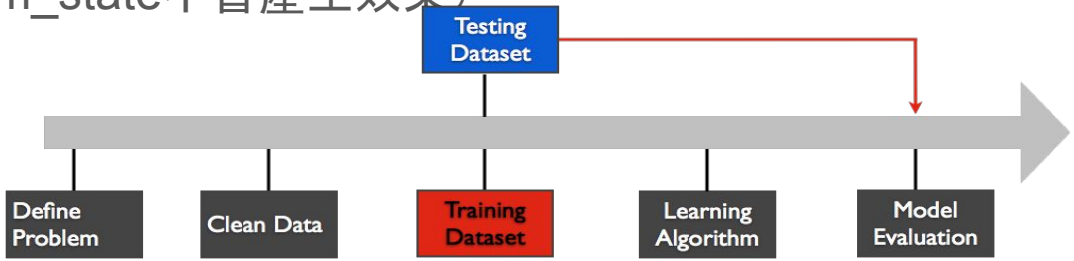
Import Library

```
from sklearn.model_selection import train_test_split
```

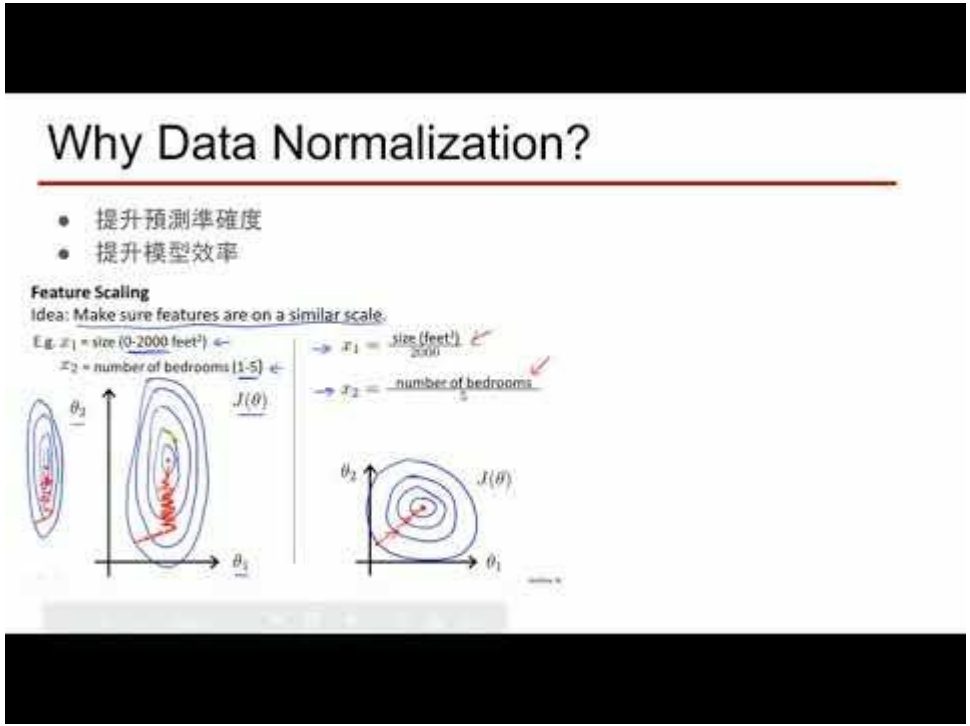
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state= 42, shuffle=False)
```



- test_size: 一般設定0.2~0.25
- shuffle: default = True（將資料順序打亂）
- random_state: 將資料打亂的方式(~random seed，如果shuffle設定為False，random_state不會產生效果)



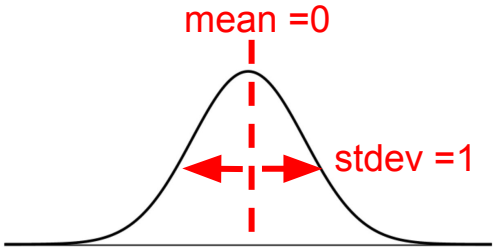
Normalization



Data Normalization

3- Data Normalization Example

- sklearn.preprocessing.scale (Z-score)
  - (X- mean)/std
  - 將資料轉換成mean為0, stdev為1的資料分佈



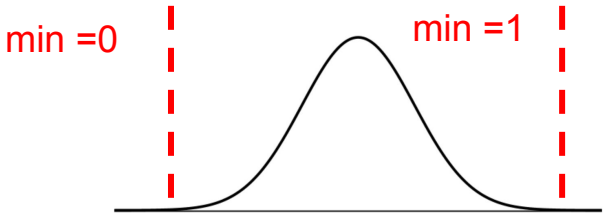

```
sklearn.preprocessing. scale (X, axis=0, with_mean=True, with_std=True, copy=True)   [source]
```

axis =0

```
[[  10.    2.7   3.6   5. ]
 [-100.    5.   -2.   10. ]
 [ 120.   20.   40.   50. ]]
```

```
[[ 0.          -0.85170713 -0.55138018 -0.82760589]
 [-1.22474487  -0.55187146 -0.852133   -0.57932412]
 [ 1.22474487   1.40357859  1.40351318  1.40693001]]
```

- sklearn.preprocessing.minmax_scale
  - (X - min(X))/(max(X) - min(X))
  - 將資料轉換成max和min轉換到feature_range



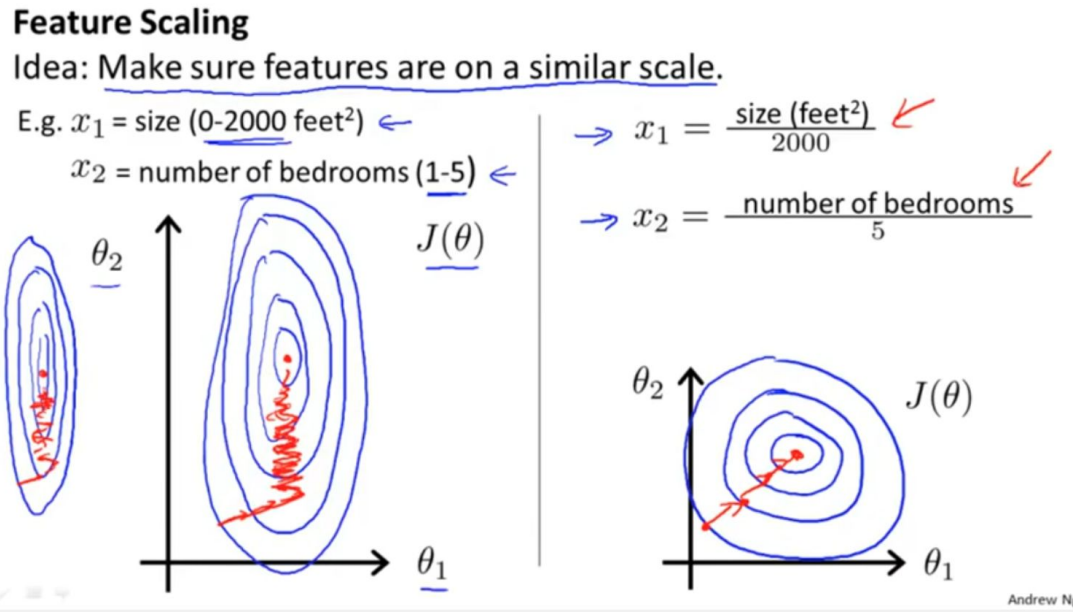
```
sklearn.preprocessing. minmax_scale (X, feature_range=(0, 1), axis=0, copy=True)   [source]
```

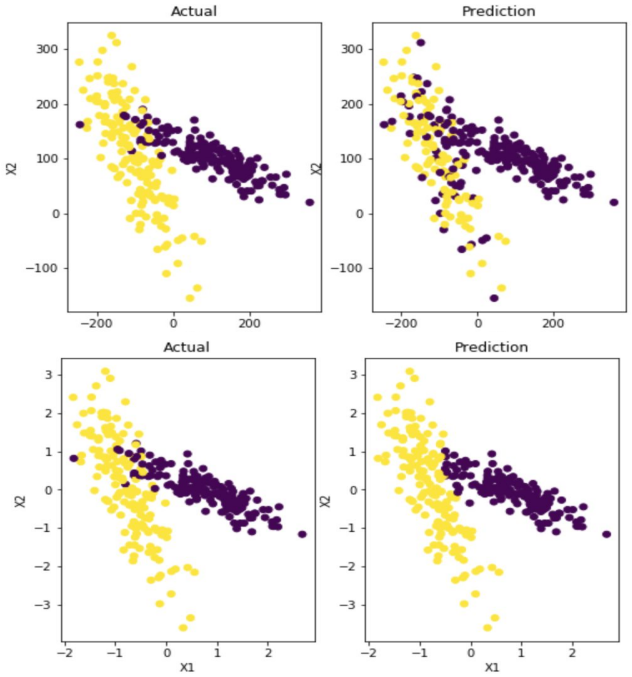
Why Data Normalization?

- 提升預測準確度
- 提升模型效率

4- Why Data Normalization Example

- 比較正規化前後的預測準確度



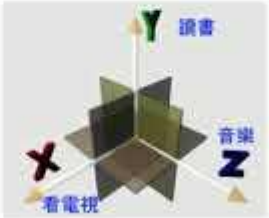


Label Encoding

One Hot Encoding（無序類別）

- 不同的興趣並沒有強度上的相關性，但regression的模型會將之視為有強度上的差異，會導致模型無法做精準的預測。

興趣- feature (X)	看電視(1)	讀書(2)	音樂(3)
成功與否- label(y)	否(0)	是(1)	否(0)



One Hot Encoding

5-One Hot Encoding Example

- 將categorical feature轉換成數字以當作模型的輸入。

ex: 最大興趣預測未來成功與否

興趣- feature（X)	看電視	讀書	音樂
成功與否- label(y)	是	否	否

```
from sklearn.preprocessing import LabelEncoder
```

	X 興趣	y 成功與否
小明	看電視	是
小林	讀書	否
小英	音樂	否
小陳	看電視	是

Label Encoder

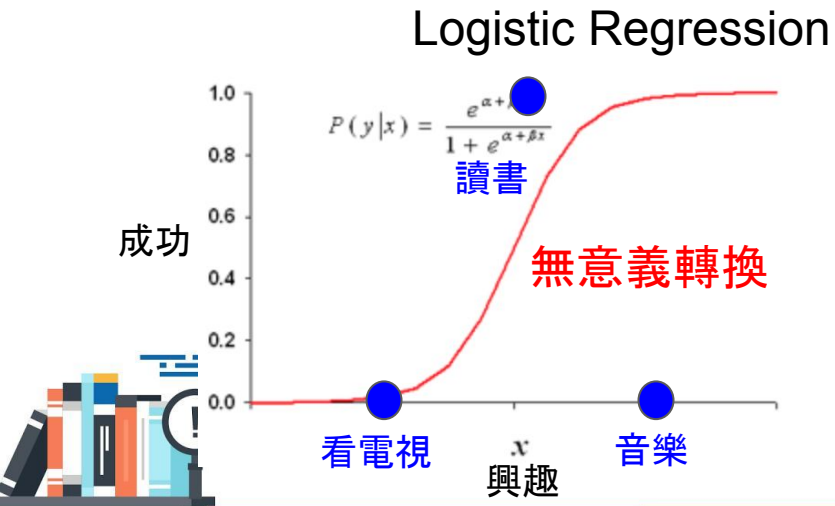
	X 興趣	y 成功與否
小明	1	1
小林	2	0
小英	3	0
小陳	1	1

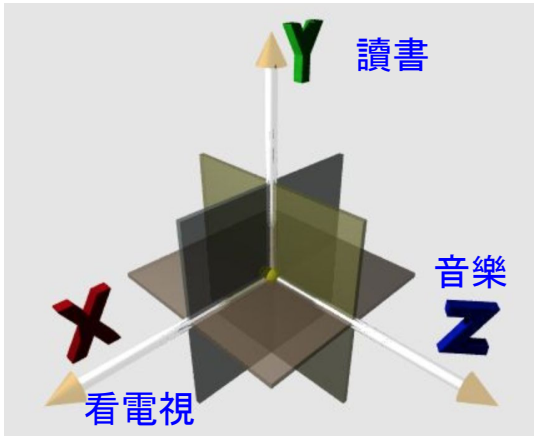
http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html#sklearn.preprocessing.LabelEncoder

One Hot Encoding（無序類別）(1/2)

- 不同的興趣並沒有強度上的相關性，但regression的模型會將之**視為有強度上的差異**，會導致模型無法做精準的預測。

興趣- feature（X）	看電視(1)	讀書(2)	音樂(3)
成功與否- label(y)	否(0)	是(1)	否(0)





One Hot Encoding（無序類別）(2/2)

6-One Hot Encoding Example

- 將無序的categorical feature轉換成數字以當作模型的輸入。

ex: 最大興趣預測未來成功與否

興趣- feature（X）	看電視	讀書	音樂
成功與否- label(y)	是	否	否

	X 興趣	y 成功與否
小明	看電視	是
小林	讀書	否
小英	音樂	否
小陳	看電視	是

```
X = pd.get_dummies(df.iloc[:,0])
X
```

One Hot Encoding

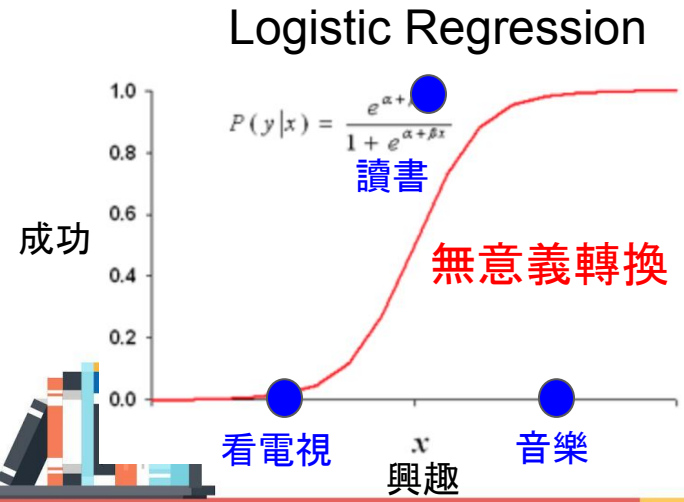
X (feature 擴充）

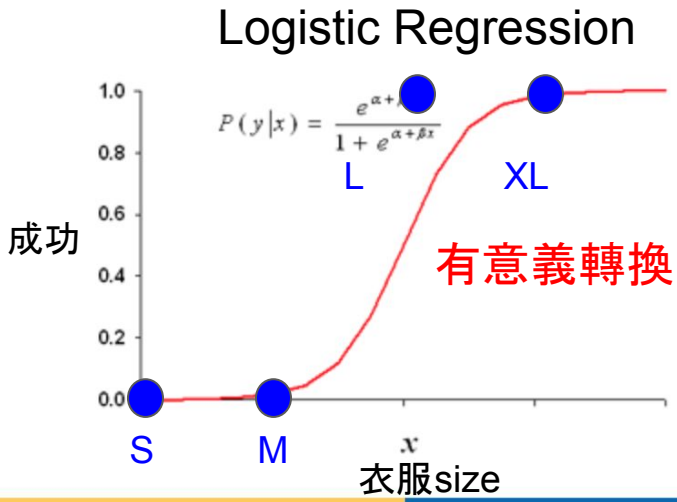
	看電視	讀書	音樂
小明	1	0	0
小林	0	1	0
小英	0	0	1
小陳	1	0	0

One Hot Encoding（有序類別）(1/2)

- 衣服size具有強度上的相關性，regression的模型會將之**視為有強度上的差異**，在意義上是正確的。

feature（X）	S(0)	M(1)	L(2)	XL(3)
成功與否- label(y)	否	否	是	是





One Hot Encoding（有序類別）(2/2)

7-One Hot Encoding Example

- 將有序的categorical feature轉換成數字以當作模型的輸入。

ex: 穿衣服的size預測未來成功與否

feature（X）	S	M	L	XL
成功與否- label(y)	否	否	是	是

```
import pandas as pd
import numpy as np
size_mapping = {'S':0, 'M':1, 'L':2, 'XL':3} #define map as dictionary
label_mapping = {'否':0, '是':1} #define map as dictionary
```

```
df_encode['衣服size'] = df_encode['衣服size'].map(size_mapping)
df_encode['成功與否'] = df_encode['成功與否'].map(label_mapping)
```

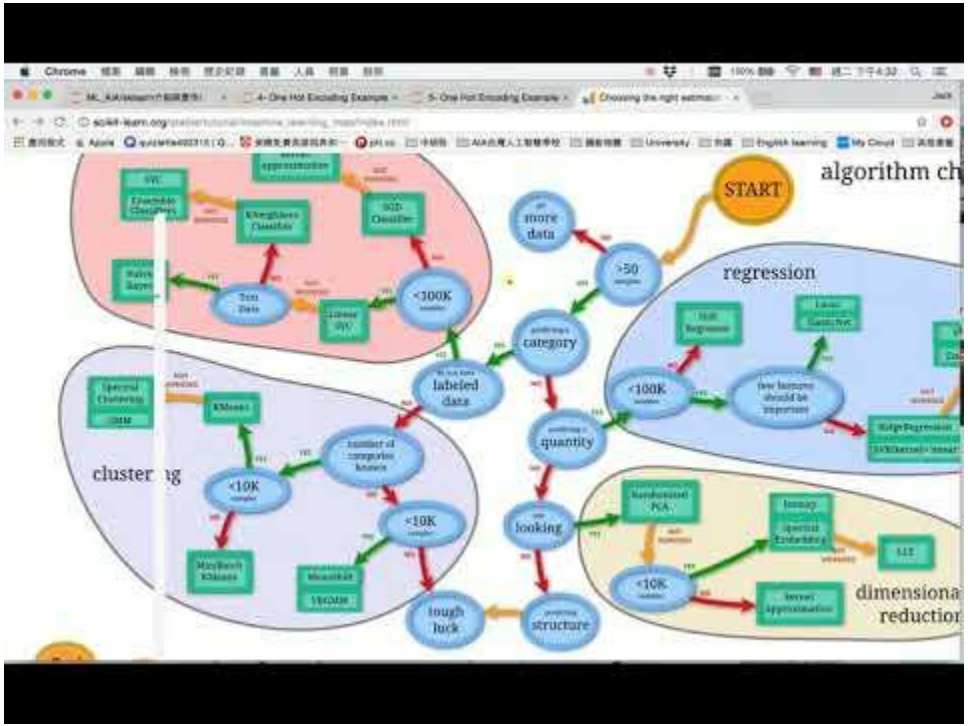
	衣服size	成功與否
小明	XL	是
小林	S	否
小英	M	否
小陳	L	是

Label Encoder

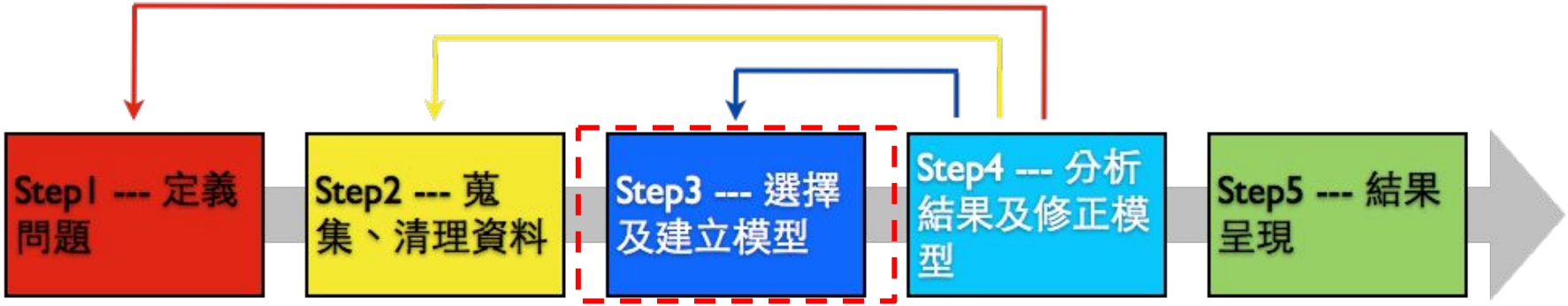
	X 衣服size	y 成功與否
小明	3	1
小林	0	0
小英	1	0
小陳	2	1

http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html#sklearn.preprocessing.LabelEncoder

Step 3 - 選擇及建立模型



Scikit Learn 機器學習模型建立的方法

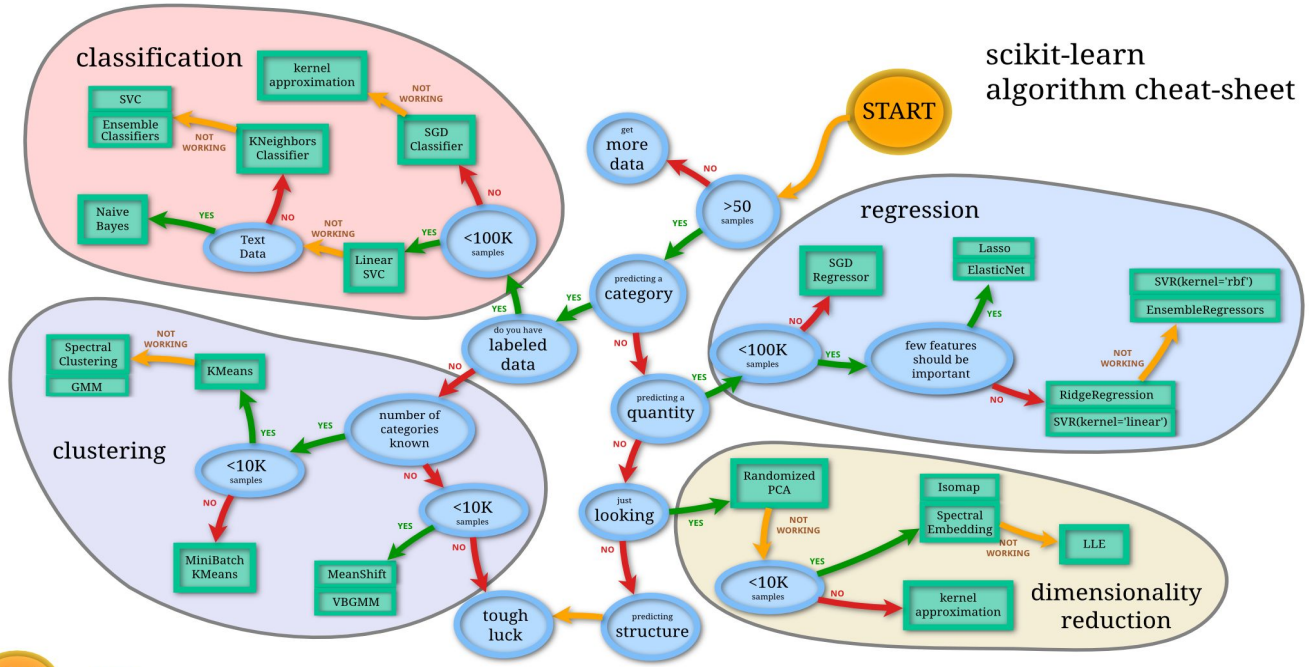


- 選擇模型 --- Machine Learning Map
- 建立模型
- 訓練模型
- 利用模型預測結果

```
# import svm classifier
from sklearn.svm import SVC
model = SVC()
model.fit(X_train, y_train)
prediction = model.predict(X)
```

Machine Learning Map

http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html



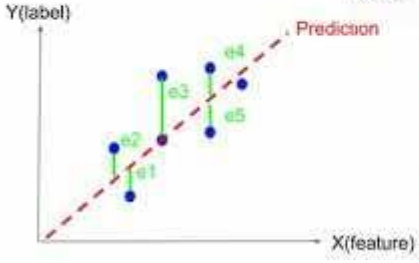
Step 4 - 分析結果及修正模型：迴歸型模型評估

Continuous

- Mean Absolute Error: $MAE = \frac{\sum_{i=1}^{N}|e_i|}{N}$
- Mean Squared Error: $MSE = \frac{\sum_{i=1}^{N} e_i^2}{N}$
- Root Mean Squared Error: $RMSE = \sqrt{\frac{\sum_{i=1}^{N} e_i^2}{N}}$

- R2 Score: coefficient of determination

$$R2 = 1 - \frac{\sum_{i=1}^{N}(e_i)^2}{\sum_{i=1}^{N}(y - y_i)^2} = 1 - \frac{total\ error}{data\ variance}$$

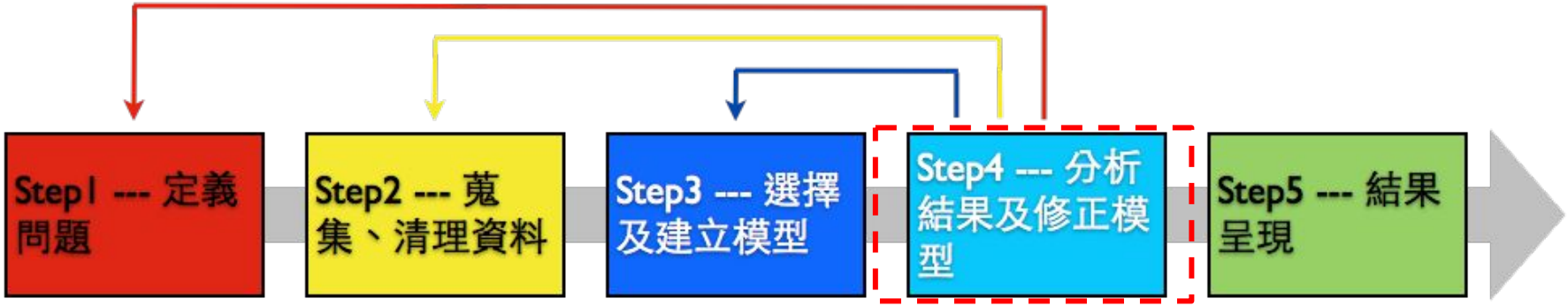


8- Model Evaluation Example

- 觀察在資料上加不同noise對預測結果造成的影響
- 了解不同metrics的意義

```
from sklearn import metrics
mae = metrics.mean_absolute_error(prediction, y)
mse = metrics.mean_squared_error(prediction, y)
r2 = metrics.r2_score(prediction, y)
```

Scikit Learn 模型評估的方法




```
from sklearn import metrics
```

- Continuous
  - Mean Absolute Error
  - Root Mean Squared Error
  - R2 Score
- Binary (Classification)
  - Accuracy
  - F1 score

迴歸型 (Regression) 模型常用評估指標

- Mean Absolute Error:
- Mean Squared Error:
- Root Mean Squared Error:

$$MAE = \frac{\sum_{i=1}^{N}|e_i|}{N}$$

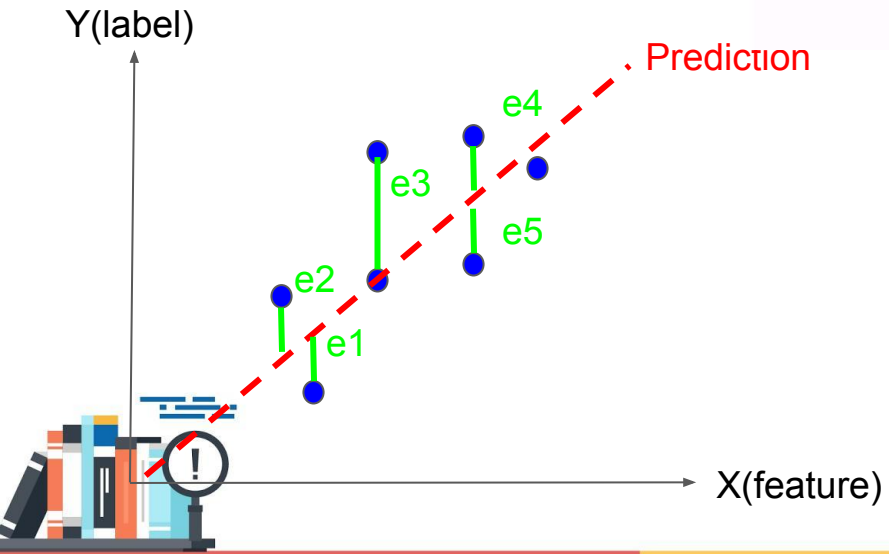
$$MSE = \frac{\sum_{i=1}^{N} e_i{}^2}{N}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} e_i{}^2}{N}}$$

8- Model Evaluation Example

- R2 Score:
  [coefficient of determination]

$$R2 = 1 - \frac{\sum_{i=1}^{N}(e_i)^2}{\sum_{i=1}^{N}(y - y_i)^2} = 1 - \frac{total\ error}{data\ variance}$$

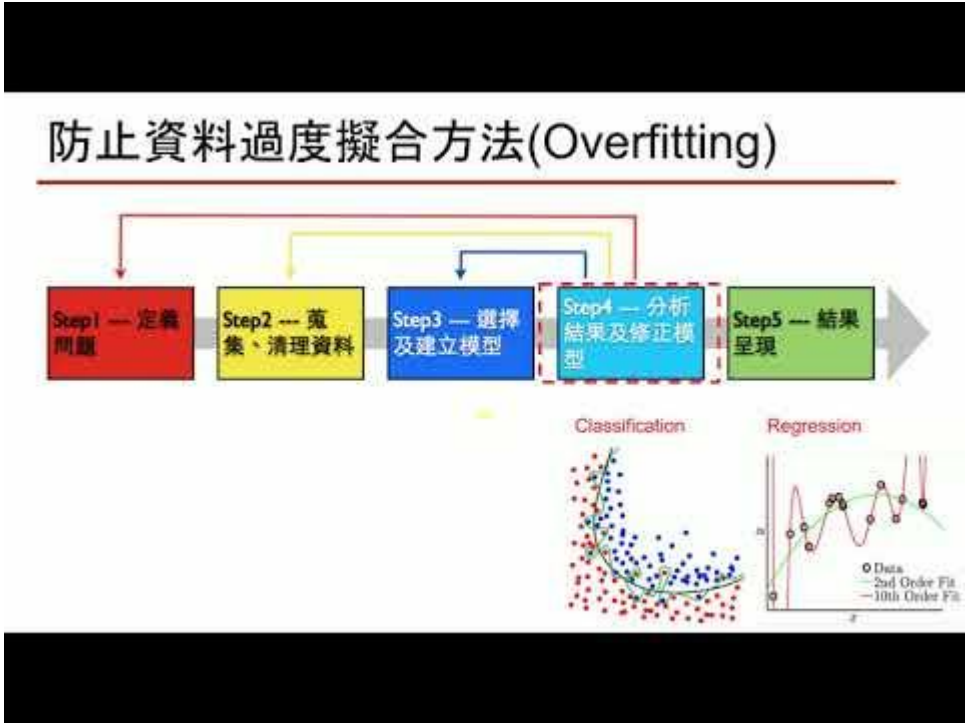


- 觀察在資料上加不同noise對預測結果造成的影響
- 了解不同metrics的意義

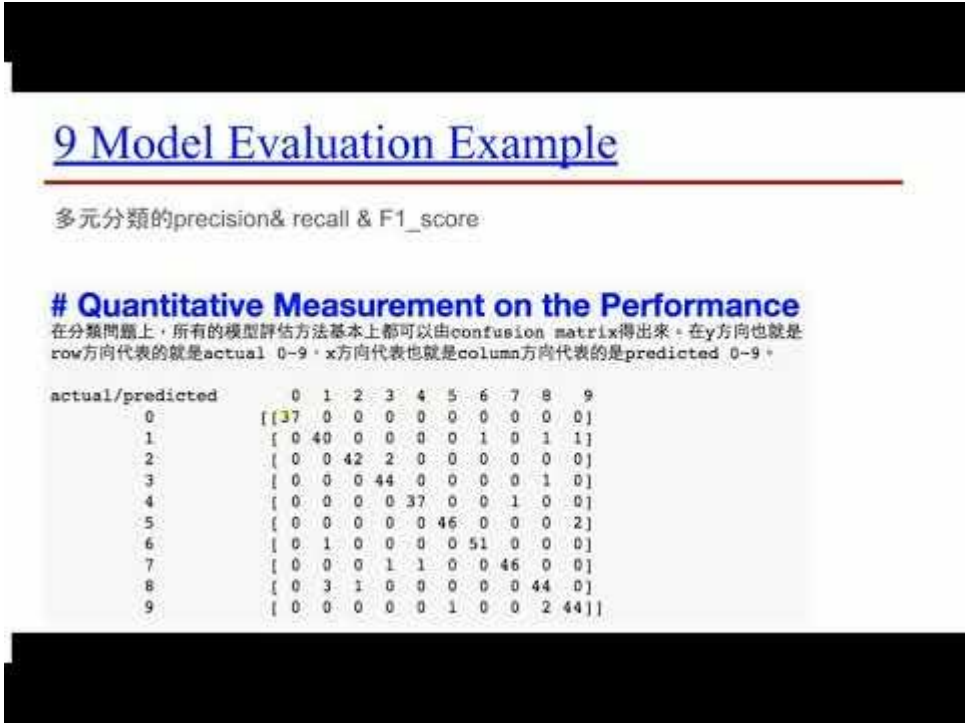
```
from sklearn import metrics

mae = metrics.mean_absolute_error(y,prediction)
mse = metrics.mean_squared_error(y,prediction)
r2 = metrics.r2_score(y,prediction)
```

防止資料過擬合



Step 4 - 分析結果及修正模型：分類型模型評估



分類型 (Classification) 常用評估指標 - Accuracy

- Accuracy:

$$\text{Accuracy} = \frac{TN+TP}{ALL= (TN+TP+FN+FP)}$$

1: 壞人
0: 好人

全猜好人可以得到高達90%的準確率呢！

樣本比例差異大

1 0 0 0 0 0 0 0 0 0

- Confusion Matrix

	Predicted 0	Predicted 1
Actual 0	True Negative (TN)	False Positive (FP)
Actual 1	False Negative (FN)	True Positive (TP)

分類型 (Classification) 常用評估指標 - F1 Score

- F1 Score: 適用於inbalanced data

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

1: 壞人
0: 好人

樣本比例差異大(imbalanced data)

1 0 0 0 0 0 0 0 0 0

- Confusion Matrix

	Predicted 0	Predicted 1
Actual 0	True Negative (TN)	False Positive (FP)
Actual 1	False Negative (FN)	True Positive (TP)

$$\text{Recall} = \frac{TP}{FN+TP} \qquad \text{Precision} = \frac{TP}{TP+FP}$$

	Recall	Precision
全說壞人(1)	100%	10%
全說好人(0)	0%	
高代表的意義	大膽、信心度低	謹慎、信心度高

分類型評估練習 / 分類型評估方式 - Confusion Matrix

9 Model Evaluation Example

- 多元分類的precision& recall & F1_score

Quantitative Measurement on the Performance

在分類問題上，所有的模型評估方法基本上都可以由confusion matrix得出來。在y方向也就是row方向代表的就是actual 0~9，x方向代表也就是column方向代表的是predicted 0~9。

```
actual/predicted      0   1   2   3   4   5   6   7   8   9
          0         [[37   0   0   0   0   0   0   0   0   0]
          1          [ 0  40   0   0   0   0   1   0   1   1]
          2          [ 0   0  42   2   0   0   0   0   0   0]
          3          [ 0   0   0  44   0   0   0   0   1   0]
          4          [ 0   0   0   0  37   0   0   1   0   0]
          5          [ 0   0   0   0   0  46   0   0   0   2]
          6          [ 0   1   0   0   0   0  51   0   0   0]
          7          [ 0   0   0   1   1   0   0  46   0   0]
          8          [ 0   3   1   0   0   0   0   0  44   0]
          9          [ 0   0   0   0   0   1   0   0   2  44]]
```

8是否分對的confusion matrix		
實際/預測	預測不為8	預測為8
實際不為8		
實際為8		

Section 1-3　[理論講授] 正則化(Regularization)