



機器學習基礎與演算法

Chapter 3 分類 (Classification)

Chapter 4 監督式學習 (Supervised Learning)

Chapter 5 參數選取 (Parameter Selection)

Chapter 6 泛化能力 (Generalization Capability)

講師投影片 課程投影片 資料與程式碼

「版權聲明頁」

本投影片已經獲得作者授權台灣人工智慧學校得以使用於教學用途,如需取得重製權以及公開傳輸權需要透過台灣人工智慧學校取得著作人同意;如果需要修改本投影片著作,則需要取得改作權;另外,如果有需要以光碟或紙本等實體的方式傳播,則需要取得人工智慧學校散佈權。

課程內容

3. 分類 (Classification)

- 3-1 [理論講授] Classification
- 3-2 [理論講授] Logistic Regression
- 3-3 [實作課程] Logistic Regression

課程內容

- 4. 監督式學習 (Supervised Learning)
 - 4-1 [理論講授] 支援向量機 (SVM)
 - 4-2 [實作課程] SVM in Scikit Learn
 - SVM Concept
 - SVC
 - SVR
 - Summary
- 5. 參數選取 (Parameter Selection)
 - 5-1 [實作課程] 調整參數
- 6. 泛化能力 (Generalization Capability)
 - 6-1[理論講授] 集成學習 (Ensemble Learning)
 - 6-2[理論講授] 泛化 (Generalization)

Code 放在Hub中的course內

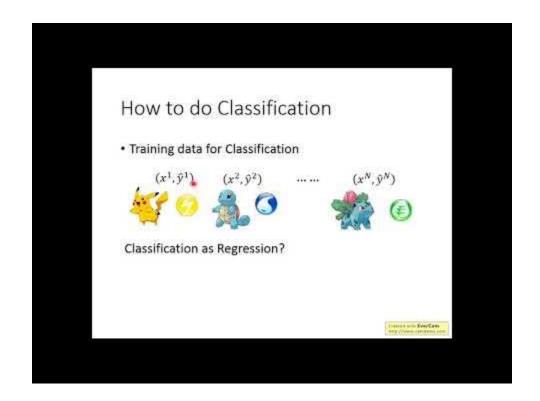
- 為維護課程資料, courses中的檔案皆為read-only, 如需修 改請cp至自身環境中
- 打開terminal, 輸入

cp -r courses-tpe/ML <存放至本機的名稱>



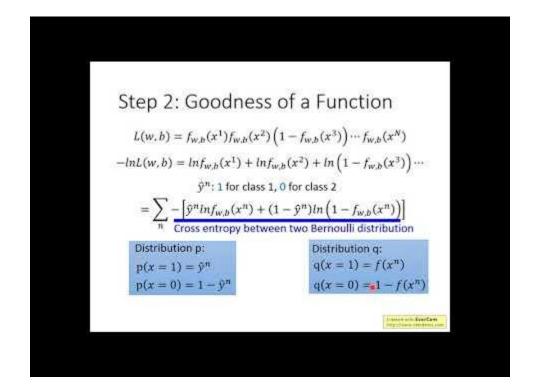
Chapter 3 分類 (Classification)

Section 3-1 [理論講授] Classification



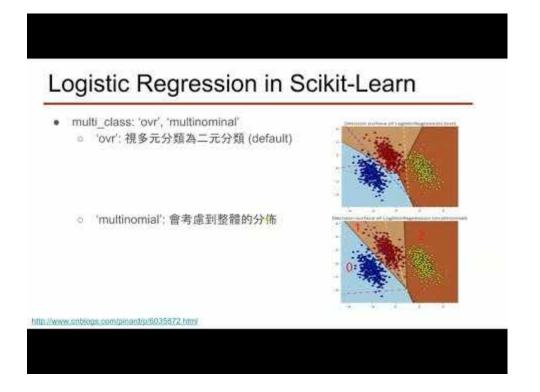


Section 3-2 [理論講授] Logistic Regression





Section 3-3 [實作課程] Logistic Regression - part 1





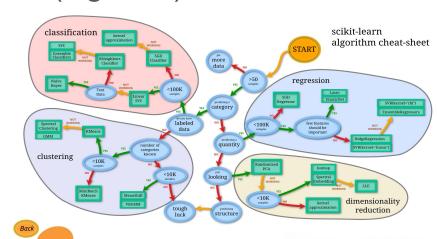
Logistic Regression

- 使用時機:
 - Label為非連續值(二元、多元分類問題)。
 - Features 和 Label之間不必有線性關係, 因為Features 的值會做non-linear的transform(sigmoid)。

$$P(y|x) = \frac{1}{1 + e^{-yw^T x}}$$

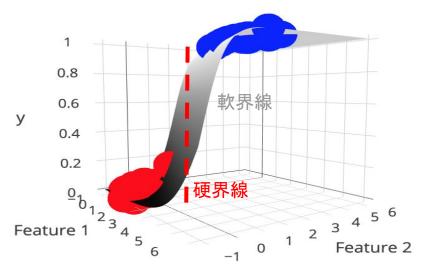
● Prediction: 0~1的機率值。





Geometric Meaning

- 利用線性關係算出分數 yw^Tx
- 將分數代進sigmoid function轉換成機率函數
- 求解最大機率



$$P(y|x) = \frac{1}{1 + e^{-yw^Tx}}$$

$$\max_{w} \prod_{i=1}^{k} P(y_i|x_i)$$

i = 1, ..., k (k training instances)



Logistic Regression in Scikit-Learn

Penalty: L1/ L2 (Lasso/Ridge)

C: default=1

Logistic Regression:
$$\min_{w} \frac{1}{2} w^T w + C \sum_{i=1}^{k} log(1 + e^{-y_i w^T x_i})$$

Linear Regression: Cost = Prediction error + $\alpha \sum$ (weights)²

Note: 和linear regression的alpha不一樣的是C值越大對weight的控制力越弱。

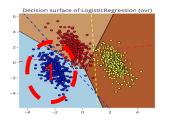
Case	Solver
Small dataset or L1 penalty	liblinear
Multinomial or large dataset	lbfgs, sag or newton-cg
Very Large dataset	sag

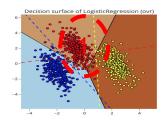
from sklearn.linear_model import LogisticRegression
model = LogisticRegression(penalty='12', dual=False,
tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1
, class_weight=None, random_state=None, solver='liblinear'
, max_iter=100, multi_class='ovr', verbose=0, warm_start=False
, n_jobs=1)



Logistic Regression in Scikit-Learn

- multi_class: 'ovr', 'multinominal'
 - 'ovr': 視多元分類為二元分類 (default)

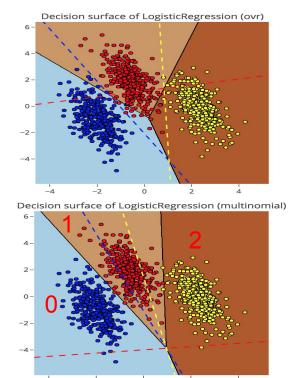






- 'multinomial': 會考慮到整體的分佈
 - 0,1 分類
 - 0,2 分類
 - 1,2 分類





Logistic Regression Example

1-Logistic Regression Example



- 調整C值觀察其對結果的影響
- o 如何解釋結果 (Hint: 準確度、擬合outlier的程度)

#調整C值

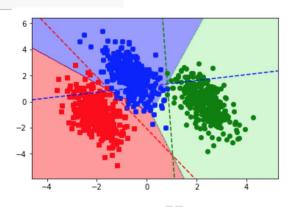
```
plot_dict = {1:231,10:232,1e2:233,1e3:234,1e4:235,1e5:236}
for i in plot_dict:
    regression example(plot_dict, i)
```

2-Logistic Regression Example



- 多元分類問題(兩類以上)
- 比較兩種multi class的方法: 'ovr', 'multinomial'





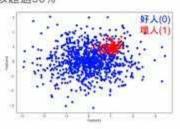
Section 3-3 [實作課程] Logistic Regression - part 2

Imbalanced Data Prediction Exercise

- 簽察抓壞人當例子:
- 動手時間 範例中總共有10%被分類為1(壞人)的類別
 - 如果全猜好人準確率就有90%, 但是這樣的做法顯然不合理
- - 用Logistic Regression 看看準確率是否可以超過90%
 - 算出Precision & Recall. 看在這樣的 狀況下抓出了多少比率壞人(Recall)? 在預測這個人是壞人時你有多少信心 (Precision)?
 - 計算F1 score

Hint 可以利用confusion matrix自行复出

如何找到最好的F1 score?





Short Summary

- 用 Logistic Regression 解二元或多元分類問題
- 利用 L1&L2 Regularization 控制模型複雜度

模型	Linear Regression	Logistic Regression
用途	連續值預測	分類問題預測
複雜度控制(L1,L2)	Alpha 越大控制力越強	C 越大控制力越弱



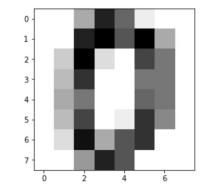
Logistic Regression Exercise動手時間



- 手寫影像為8x8的灰階影像
- 將每個影像展平成64個features的dataset來做預測
- 學習點
 - 調整C值及multi_class、solver
 - 用accuracy來評估模型結果
 - 利用confusion matrix來分析 哪些數字較容易分辨哪些較 不容易

plt.imshow(digits.images[0], cmap=plt.cm.binary, interpolation='nearest')

<matplotlib.image.AxesImage at 0x116b34e80>



```
array([[ 0., 0., 5., 13., 9., 1., 0., 0.],
            [ 0., 0., 13., 15., 10., 15., 5., 0.],
            [ 0., 3., 15., 2., 0., 11., 8., 0.],
            [ 0., 4., 12., 0., 0., 8., 8., 0.],
            [ 0., 5., 8., 0., 0., 9., 8., 0.],
            [ 0., 4., 11., 0., 1., 12., 7., 0.],
            [ 0., 2., 14., 5., 10., 12., 0., 0.],
            [ 0., 0., 6., 13., 10., 0., 0., 0.]]
```



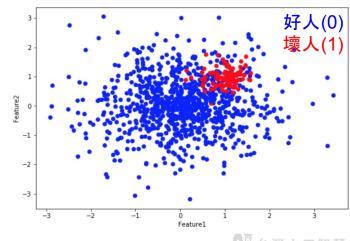
Imbalanced Data Prediction Exercise 動手時間



- 警察抓壞人當例子:
 - 範例中總共有10%被分類為1(壞人)的類別
 - 如果全猜好人準確率就有90%,但是這樣的做法顯然不合理
- 提示:
 - 用Logistic Regression 看看準確率是否可以超過90%
 - 算出Precision & Recall, 看在這樣的 狀況下抓出了多少比率壞人(Recall)? 在預測這個人是壞人時你有多少信心 (Precision)?
 - 計算F1_score

Hint:可以利用confusion matrix自行算出

如何找到最好的F1 score?



Classification and Regression Exercise 動手時間

Exercise1- Classification ippyier



- 選擇不同的kernel. 並調整參數。
- 利用Accuracy評估預測結果
- 利用F1 score和confusion matrix評估預測結果

Exercise2- Regression in Exercise2- Regression



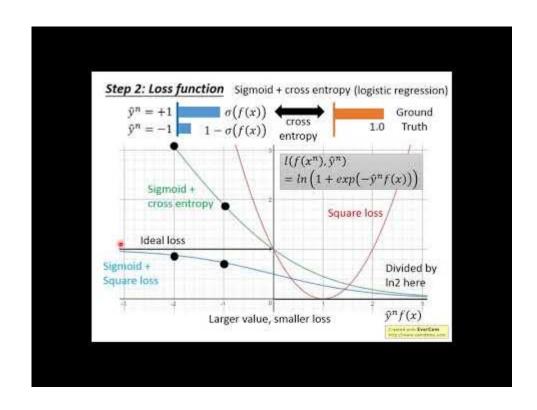
- 觀察各Feature和房價之間的關係並選擇適當的kernel做預測
- 不生成新Feature的情況下是否可以將讓預測結果接近先前做Linear Regression的結果



Chapter 4 監督式學習 (Supervised learning)

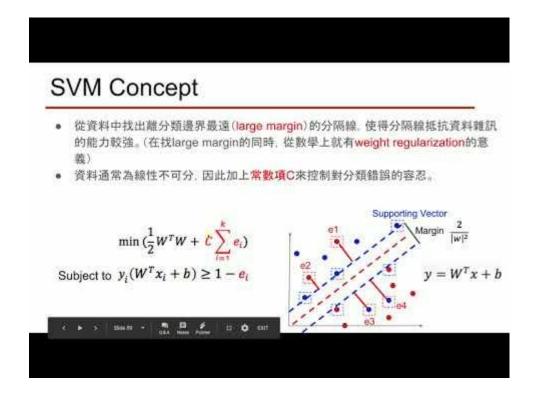
- 範例程式(example)的檔名會以藍色字體顯示且旁邊附上
- 練習(exercise)的檔案以紅色字體顯示且旁邊附上

Section 4-1 [理論講授] 支援向量機 (SVM)





Section 4-2 [實作課程] SVM in Scikit Learn - SVM Concept





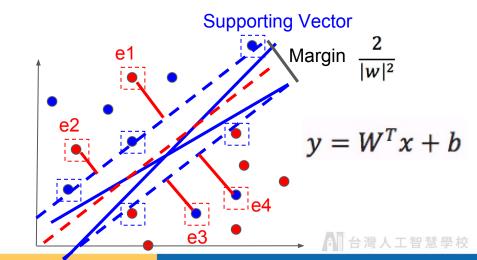
SVM in Scikit Learn - SVM Concept

- 從資料中找出離分類邊界最遠(large margin)的分隔線, 使得分隔線抵抗資料雜訊的能力較強。(在找large margin的同時, 從數學上就有weight regularization的意義)
- 資料通常為線性不可分, 因此加上常數項C來控制對分類錯誤的容忍。

$$\min\left(\frac{1}{2}W^TW + C\sum_{i=1}^k e_i\right)$$

Subject to $y_i(W^Tx_i + b) \ge 1 - e_i$

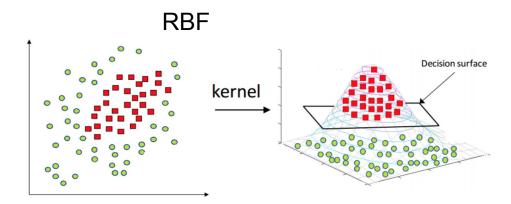




SVM Kernel Trick

- 一般狀況下,資料多為線性不可分。藉由kernel可以將資料做非線性的轉換到多維空間,甚至無限多維空間,使得SVM可以在此將資料做分類。
- Kernel即是將資料做空間轉換,代替linear regression手動生feature的方式
- 一般常見的Kernel方式:
 - Polynomial:高次方轉換
 - Radial Basis Function: 高斯轉換 ⇒ 無限多維空間

Polynomial



SVM in Scikit Learn - SVC





SVM in Scikit Learn

- C控制對錯誤的容忍度, degree、gamma控制kernel的複雜度。
- probability = True讓model可以output
- SVM 在Scikit Learn中做multi classification的演算法為 'ovr'。

```
from sklearn import svm
model = svm.SVC(
# {linear, poly, rbf}
kernel = 'linear',
# used only if kernel='poly'
degree = 3,
# used only if kernal='rbf'
gamma = 0.7
# enable predict_proba()
probability = True,
# smaller values, stronger regularization
C = 1.0
model.fit(X,y)
SVC(C=1.0, cache size=200, class weight=None, coef0=0.0,
 decision function shape='ovr', degree=3, gamma=0.7, kernel='linear',
  max iter=-1, probability=True, random state=None, shrinking=True,
  tol=0.001, verbose=False)
```

SVM Probability Output:

SVM Example

- 1- SVM Kernel Comparison Example---iris
- - 比較不同Kernel產生邊界的形狀差異
- 2- SVM C Parameters Example---iris



- 調整C在三個不同的Kernel上對模型複雜度的限制
- C越大邊界越複雜/簡單?
- 3- SVM Other Parameters Example---iris



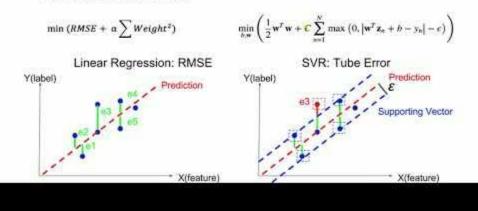
- 調整degree/ gamma
- RBF Kernel 是否可以做到極高準確率?



SVM in Scikit Learn - SVR

SVM for Linear Regression (SVR)

概念與SVM分類問題類似、找出一條線位於資料Supporting Vector的中心、並用常數C來控制對錯誤的容忍。





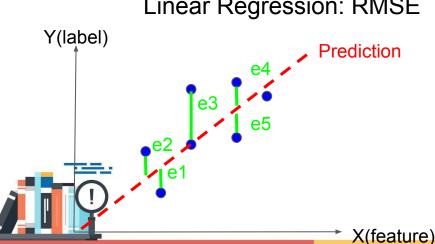
SVM for Linear Regression (SVR)

概念與SVM分類問題類似,找出一條線位於資料Supporting Vector的中心,並用 常數C來控制對錯誤的容忍。

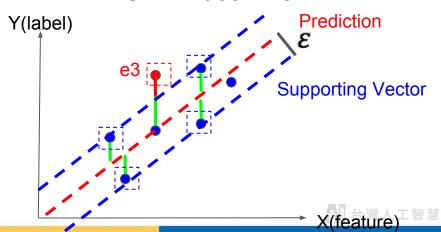
$$\min (RMSE + \alpha \sum Weight^2)$$

$$\min_{b,\mathbf{w}} \left(\frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{n=1}^{N} \max \left(0, \left| \mathbf{w}^T \mathbf{z}_n + b - y_n \right| - \epsilon \right) \right)$$

Linear Regression: RMSE



SVR: Tube Error



SVR in Scikit Learn

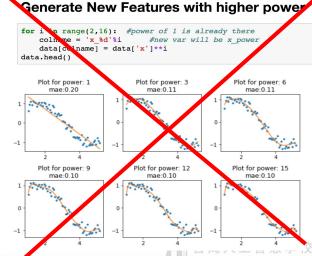
- 同SVM, 常數C用來控制模型對錯誤的容忍度。
- kernel有'linear', 'poly' 和 'rbf' 可以選擇。
 - 'poly': 用degree來控制模型複雜度(高次項擴展)
 - 'rbf':用gamma來控制模型複雜度(高斯轉換)

```
from sklearn.svm import SVR
model = SVR(kernel='poly', degree=1, C=0.01)
model.fit(X_train,y_train)
```

4- SVR Example jupyter

- 調整C, degree, gamma看看對預測結果的影響。
- 和之前提到的Linear Regression比較, SVR並不需要自行做兩種轉換生成 Footure, 可利用Kernel将

要自行做座標轉換生成Feature, 可利用Kernel將 ②資料自動擴展到多維空間。



SVM in Scikit Learn - SVM Short Summary

Short Summary

- 用 SVM 解二元、多元分類及回歸問題
- 用不同Kernal將資料做空間轉換解非線性問題
- 調整degree或gamma來增加模型複雜度
- 利用C做控制模型對誤差的容忍度(模型複雜度)

模型	Linear Regression	Logistic Regression	SVM
用途	連續值預測	分類問題預測	連續值預測問題 分類問題預測
複雜度控制(L1,L2)	Alpha 越大控制力越強	C越大控制力越弱	C越大控制力越弱



Short Summary

- 用 SVM 解二元、多元分類及回歸問題
- 用不同Kernel將資料做空間轉換解非線性問題
- 調整degree或gamma來增加模型複雜度
- 利用C做控制模型對誤差的容忍度(模型複雜度)

模型	Linear Regression	Logistic Regression	SVM
用途	連續值預測	分類問題預測	連續值預測問題 分類問題預測
複雜度控制(L1,L2)	Alpha 越大控制力越強	C 越大控制力越弱	C 越大控制力越弱

SVM&SVR Exercise動手時間

SVM Exercise1 - Classification (upyter



- 選擇不同的kernel. 並調整參數。
- 利用Accuracy評估預測結果
- 利用F1 score和confusion matrix評估預測結果

SVR Exercise2 - Regression in incomplete statement with the second secon

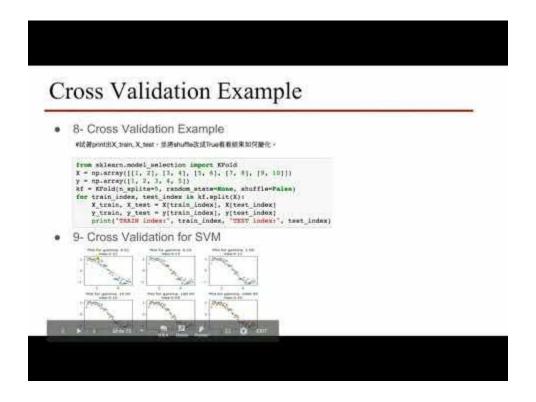


- 觀察各Feature和房價之間的關係並選擇適當的kernel做預測
- 不生成新Feature的情況下是否可以將讓預測結果接近先前做Linear Regression的結果



Chapter 5 參數選取 (Parameter Selection)

Section 5-1 [實作課程] 調整參數





選擇模型參數 - 評估模型好壞

Training Testing

 k-fold Cross Validation: 將其中一份validation set從training data抽出, 剩下資料 拿來做訓練, 如此重複k次, 並將結果平均, 來得到更robust的評估結果。

EX: 5- fold CV

	Training				
-	Validation	Validation	Validation	Validation	Validation
	1	2	3	4	5

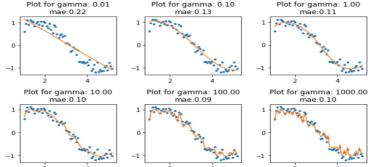
Cross Validation Example



#試著print出X train, X test, 並將shuffle改成True看看結果如何變化。

```
from sklearn.model selection import KFold
X = np.array([[1, 2], [3, 4], [5, 6], [7, 8], [9, 10]])
y = np.array([1, 2, 3, 4, 5])
kf = KFold(n splits=5, random state=None, shuffle=False)
for train index, test index in kf.split(X):
    X train, X test = X[train index], X[test index]
    y train, y test = y[train index], y[test index]
    print("TRAIN index:", train index, "TEST index:", test index)
```

6- Cross Validation for SVM





Grid Search Cross Validation

選擇模型參數 -Grid Search CV 10- Cross Validation Example—iris:比較選擇參數前後的準確度及output最佳參數。 Grid Search for SVM Parameters Erose Milearn, model_selection import GridSearchCV parameters= ("Keroel"+|"Linear", "rof"), "C"+|0-01,0-1,1,101; "gamma"+|0-05,0-1,1,101) podel = svm.8VC() heat_model = GridSearchCV(model, parametern, cp=5, scoring="acquiracy") heat_model.fit(X, y) • GridSearchCV input: • 欲運行的模型 • 欲scan的參數並將之存成dict的形式 • cv為做cross validation的folds • scoring可指定sklearn內理的eval metrics

http://scikit-leam.org/stable/modules/generated/skleam.model_selection.GndSearchCV.html



選擇模型參數 -Grid Search CV



7- Cross Validation Example---iris:比較選擇參數前後的準確度及 output最佳參數。

Grid Search for SVM Parameters

```
from sklearn.model selection import GridSearchCV
parameters= { 'kernel': ['linear', 'rbf'], 'C': [0.01,0.1,1,10], 'gamma': [0.01,0.1,1,10]}
model = svm.SVC()
best model = GridSearchCV(model, parameters, cv=5, scoring='accuracy')
best model.fit(X, y)
```

- GridSearchCV input:
 - 欲運行的模型
 - 欲scan的參數並將之存成dict的形式
 - cv為做cross validation的folds
 - scoring可指定sklearn內建的<u>eval metrics</u>

Gamma/C	0.01	0.1	1	10	
0.01					
0.1					
1					
10					

n.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

Machine Learning Practice

- Wine Quality
 - Task1: Classify red/ white wine (目標: Accuracy)
 - Task2: Predict wine quality(目標:F1 Score)
- Pima Indians Diabetes (目標:F1_Score)
 - Datasets
- <u>German Credit Data</u>(目標:F1_Score)
 - Datasets

Note:

- 點連結即可進到題目 說明網址
- 第一題datasets已經先幫大家整理 好放在資料夾內, 二三題請自行從 Datasets連結導入。

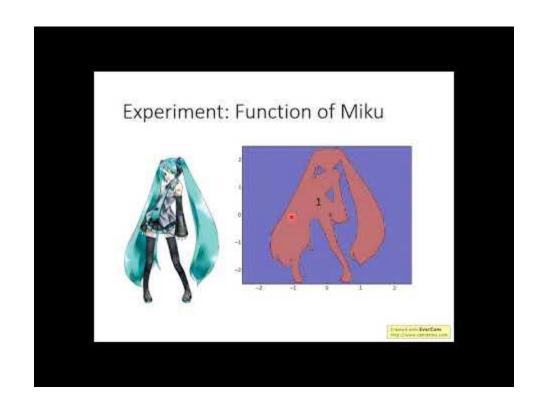


Hint: 試著畫圖了解資料

from UCI Machine Learning Repository

Chapter 6 泛化能力 (Generalization Capability)

Section 6-1 [理論講授] 集成學習 (Ensemble Learning)





Section 6-2 [理論講授] 泛化 (Generalization)

