

# 線性迴歸(Linear Regression)

# 學習目標

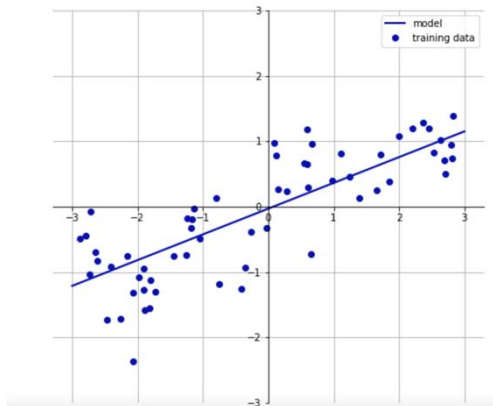
- 線性迴歸原理
- 使用預測房價公開資料集
- 以線性迴歸模型預測房價

# 線性迴歸原理

- 什麼是迴歸學習
- 什麼是線性迴歸
- 簡單線性迴歸
- 多元線性迴歸

# 什麼是迴歸學習

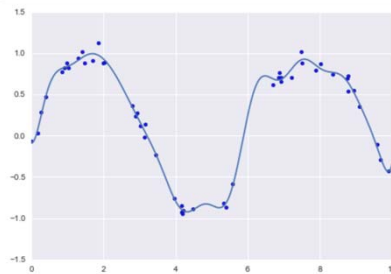
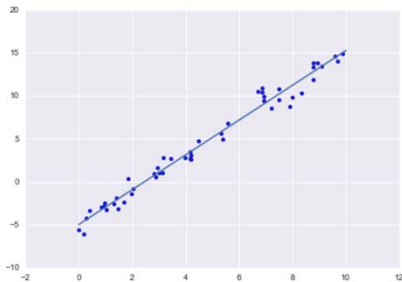
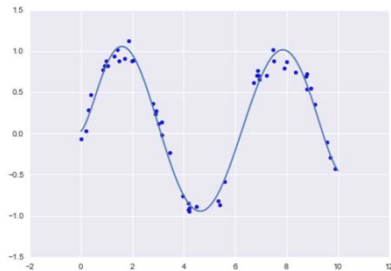
- 迴歸學習是一種近似方法，從未知機率的分布的隨機樣本中獲得目標函數，例如從大資料中找出一個規則，像是股票預測。
- 對於存在統計關係的變數，透過大量試驗來獲得的統計資料來建置目標函數去逼近該關係，即是迴歸學習



此例希望由大量的藍色資料離散點，經由重複的學習來取得連續藍色線

# 什麼是線性迴歸

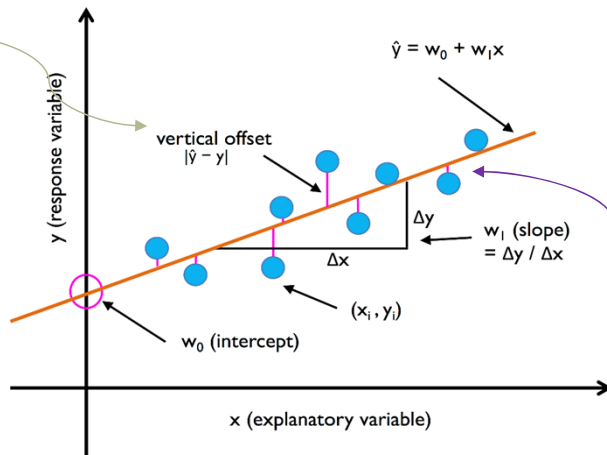
- 線性迴歸模型是學習迴歸工作的好起點
- 以線性表示，簡單明瞭
- 監督式機器學習的子類別
- 因為擬合非常快速，這個模型很受歡迎
- 迴歸分析在預測連續輸出的值



# 簡單線性迴歸

- 簡單線性迴歸目標就是擬合一條線到資料(樣本點)
- 一條線直線擬合是一個型式為 $\hat{y} = \omega_1 x + \omega_0$ 的模型， $\omega_1$ 是斜率(slope)， $\omega_0$ 是截距(intercept)

偏移值(offsets)或殘差(residuals)



線性迴歸就是希望  
找到的資料都可以fit直線

# 多元線性迴歸表示法

- 多元線性迴歸就是簡單線性迴歸的延展

$$y = w_0x_0 + w_1x_1 + \cdots + w_mx_m = \sum_{i=0}^m w_ix_i = w^T x$$

這裏的 $w_0$   
是當 $x_0 = 1$ ，  
其他 $x_i = 0$   
時的截距

有 $m$ 個 $x$

# 使用預測房價公開資料集

- 探索房屋的數據集
- 探索式數據分析
- 視覺化數據集中的重要特徵
- 相關矩陣(混淆矩陣)



# 探索房屋的數據集

- 使用pandas的read\_csv函數
- 波士頓房屋數據集包含506個樣本，包含14個特徵

	犯罪率													房價
	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222.0	18.7	396.90	5.33	36.2

## 也可以直接從Scikit-Learn匯入Dataset

- `from sklearn.datasets import load_boston`
- `boston_dataset = load_boston()`
- `boston_dataset.keys()` # 查看此資料集的Keys
- `# output: dict_keys(['data', 'target', 'feature_names', 'DESCR', 'filename'])`
- `data` : 每個房子的資訊
- `target` : 每個房子的價格
- `feature_names` : 每個房子的特徵
- `DESCR` : 這個資料集的描述
- `filename` : 此資料集的檔案位置

# 把Scikit-Learn的資料轉成Pandas.DataFrame

- `boston_df=pd.DataFrame(boston_dataset['data'],  
                              columns=boston_dataset['feature_names'])`
- `boston_df['target']=boston_dataset['target' ]`
- `boston_df.head()`

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	target
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33	36.2

# 探索式數據分析

- 使用探索式數據分析(Exploratory Data Analysis, EDA)工具
- 運用視覺化的基本的統計等工具，來看資料
- 安裝seaborn套件

```
pip install seaborn
```

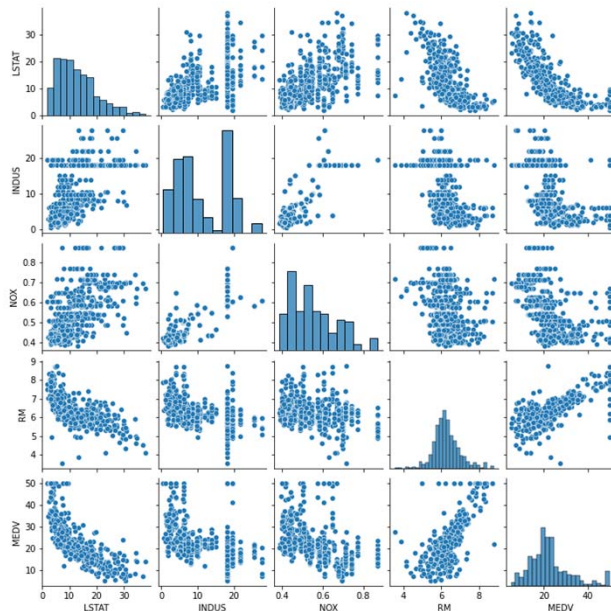
- 安裝Matplotlib套件

```
pip install matplotlib
```

## ■ 使用探索式數據分析(Exploratory Data Analysis · EDA)工具

### □ 散點圖矩陣

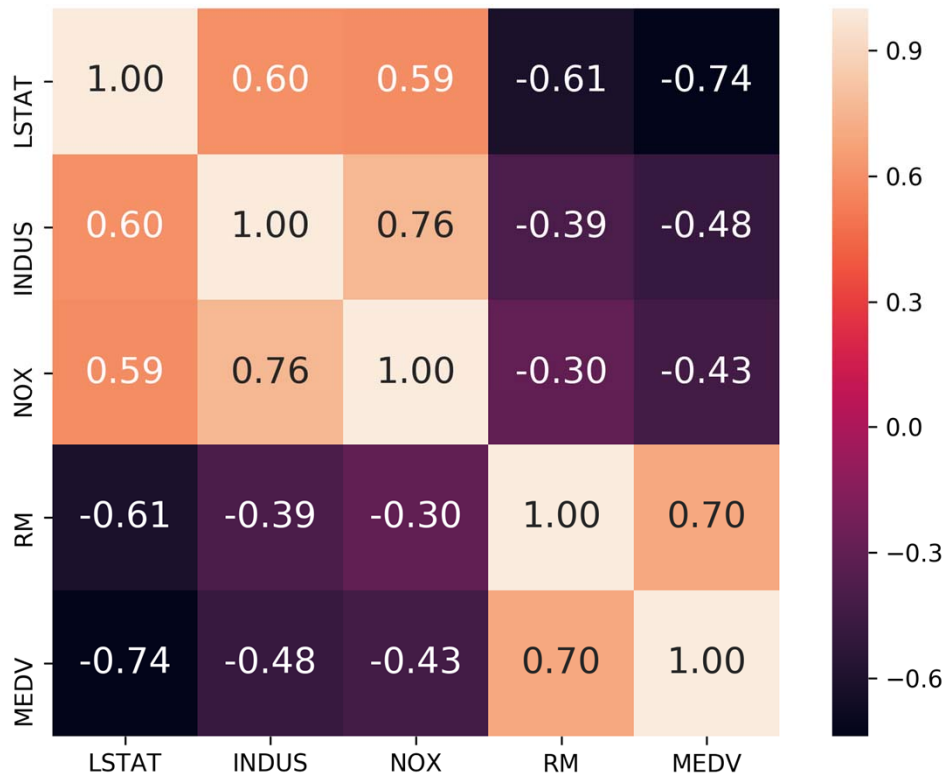
- `cols = ['LSTAT', 'INDUS', 'NOX', 'RM', 'MEDV']`
- `sns.pairplot(df[cols], size=2)`
- `plt.tight_layout()`
- `plt.show()`



# 相關矩陣

```
import numpy as np
cm = np.corrcoef(df[cols].values.T)
#sns.set(font_scale=1.5)
hm = sns.heatmap(cm,
                  cbar=True,
                  annot=True,
                  square=True,
                  fmt='.2f',
                  annot_kws={'size': 15},
                  yticklabels=cols,
                  xticklabels=cols)

plt.tight_layout()
plt.show()
```



# 以線性迴歸模型預測房價

- 建立簡單線性迴歸流程
- 建立迴歸預測模型
- 迴歸學習
- 檢驗結果

# 建立簡單線性迴歸流程

1. 依據預測目標，確立自變數(x)跟因變數(y)
2. 建立迴歸預測模型，例如： $y = ax + b$
3. 進行相關分析，例如用什麼誤差法，怎麼更新參數
4. 檢驗迴歸預測模型，計算預測誤差
5. 計算並確定預測值

Scikit-learn使用LinearRegression實現簡單線性迴歸學習

```
from sklearn.linear_model import LinearRegression  
model = LinearRegression(fit_intercept=True)
```



# 建立迴歸預測模型

## ■ 用波士頓房屋數據集

□ RM ( 每套住房的平均房間數 ) : 自變數 ( x )

□ MEDV ( 房價 ) : 因變數 ( y )


迴歸預測模型： $y = ax + b$

房間數

房價

# 迴歸學習

- Scikit-learn的LinearRegression套件



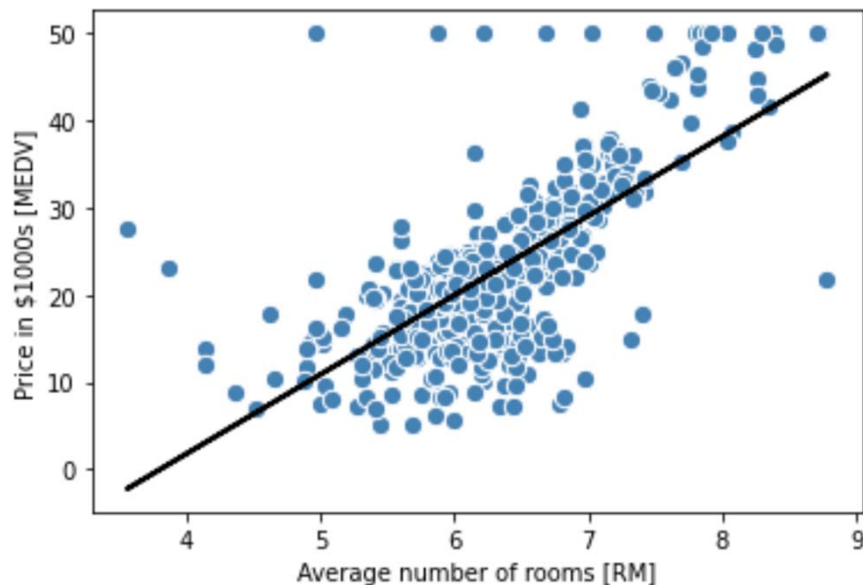
```
In [15]: slr = LinearRegression()
slr.fit(X, y)
y_pred = slr.predict(X)
print('Slope: %.3f' % slr.coef_[0])
print('Intercept: %.3f' % slr.intercept_)
```

Slope: 9.102

Intercept: -34.671

# 檢驗結果

- 圖中藍色散點資料對應X軸 ( RM ) 跟Y軸房價 ( MEDV )
- 黑線是線性迴歸模型



# 重點精華回顧

- 學習簡單線性迴歸
- 使用公開資料集及資料圖形化分析
- 建立線性迴歸模型
- 執行線性迴歸學習
- 預測結果圖形化

# 程式練習題

## ■ 多項式迴歸練習

多項式迴歸也是線性模型，在scikit-learn的實作上，是用PolynomialFeature轉換器投影到高維

## ■ 簡單線性迴歸練習

使用scikit-learn機器學習套件從隨機產生的資料，找出簡單線性函數的斜率(a)跟截距(b)

## ■ 多元線性迴歸練習

此練習題要用Scipy語法的來建立一個5X5的稀疏矩陣