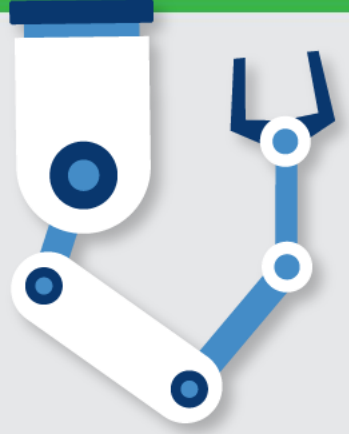
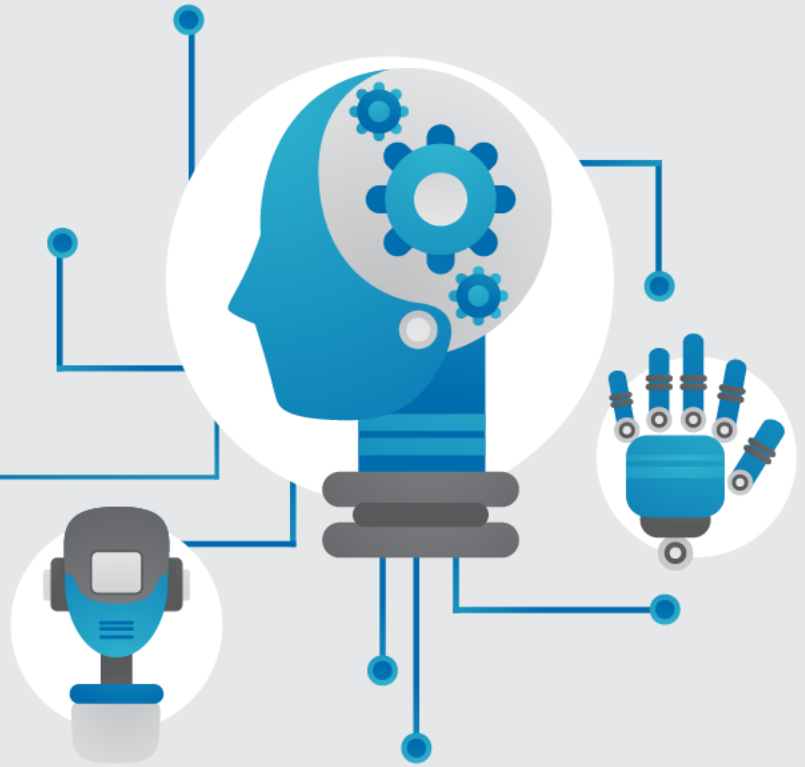


數據資料前處理



數據資料前處理

- 特徵選擇
- 資料清理
  - 處理資料缺漏
  - 轉換資料非數值類型
  - 刪除重複資料
  - 挑出離群值
  - 處理類別不平均
- 加快速度、提高精準
  - 標準化
  - 正規化
- 訓練策略
  - 回歸、分類

特徵選擇

評估特徵重要性

- 使用「羅吉斯迴歸」

過濾掉與預測目標不相關的特徵

- 使用pandas DataFrame的drop刪除欄位
- 範例程式：

```
import pandas as pd
# 刪除不需要的欄位
df = pd.read_csv('data.csv')
df.drop(['id', 'name'])
```

處理資料缺漏

當欄位資料缺漏時

- 刪除該特徵
  - ✓ 特徵的資料缺漏太多
  - ✓ 使用pandas DataFrame的drop直接刪除該特徵之欄位
- 填補缺漏處
  - ✓ 如果只有少數項特徵欄位有缺漏，可以選擇用填補的方式處理，填補的方法通常類別型的補0，數值型的補平均數或中位數
  - ✓ 使用pandas Series的median(),mean()和fillna()填補
  - ✓ 範例程式

```
mean=data['price'].mean()
data['price']=data['price'].fillna(mean)
```

轉換資料非數值類型

- 像是地址、單字等等，需要轉換成數值資料。
  - 地址：可以轉成經緯度
  - 單字：可以把會出現的單字排成序列，轉成對應的索引值
    - ✓ 使用Series.map
    - ✓ 範例程式

```
data['level']=data['level'].map({'normal': 0, 'mild': 1, 'moderate': 2, 'severe': 3})
```

- 類別類型的數據，無大小關係，轉成one-hot encoding

刪除重複資料

- **正常情況下，重複的資料需要被刪除，**

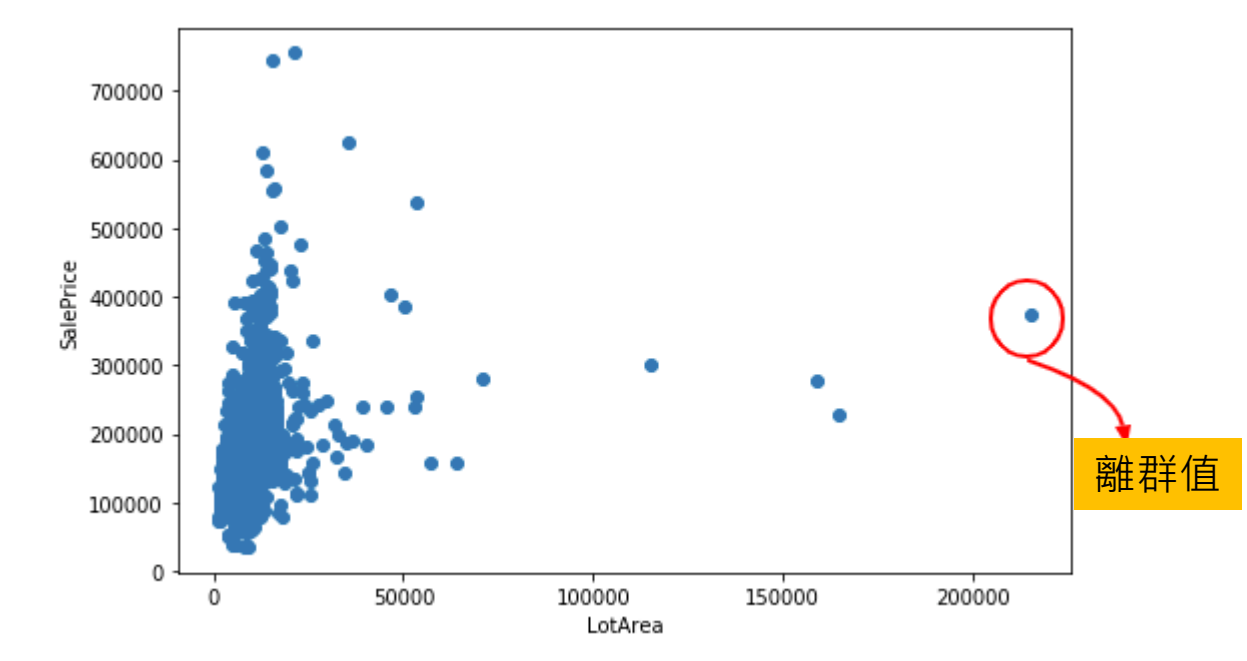
  除非是某些數據增量的策略產生的重複資料。

  - 使用pandas DataFrame.drop_duplicates刪除重複欄位
  - 範例程式

```
data=data.drop_duplicates(keep='first')
```

挑出離群值

- 藉由資料視覺化工具，可以發現某些離群值有可能是特例，需要特別挑出來判斷，加入處理會影響數據正規化等等。



處理類別不平均

- 類別不平均可能會讓模型訓練偏頗，運算的資料大多屬於某個類別，會造成overfitting，模型效率不好。

- 如不能取得更多資料量，可複製數量較少的類別，讓每個batch取得的資料類型較平均，並加上class_weight，讓loss計算可以平均一點。

- 或者將量多的類別進行分群，僅挑出比較有差異的加入訓練，使數量減少。

處理類別不平均

- 使用類別權重範例程式

```
from sklearn.utils import class_weight
import numpy as np
y = [1,1,1,1,0,0,1]
cw=class_weight.compute_class_weight('balanced', np.unique(y), y)
print(cw)
```

```
In [17]: print(cw)
[1.75 0.7 ]
```

標準化

- 如果各特徵的範圍差異很大，或單位不一樣，有可能造成某個特徵決定整個模型的預測，就需要做資料標準化，通常使用 z-score（或稱**標準化值**）。

- $z = (x - \mu) / \sigma$
  - x : 需要被標準化的原始分數
  - μ: 母體的平均值
  - σ : 母體的標準差

標準化

範例程式

```
import pandas as pd
import numpy.random as rand

df = pd.DataFrame(rand.randint(100,200,size=(5, 3)),
columns=['A','B','C'])
print(df)

df_zscore = (df - df.mean())/df.std()
print(df_zscore)
```

```
    A    B    C
0  118  149  199
1  166  115  187
2  100  197  175
3  182  175  176
4  179  191  104
```

```
          A         B         C
0 -0.825567 -0.485982  0.828298
1  0.452730 -1.493505  0.505585
2 -1.304928  0.936404  0.182871
3  0.878829  0.284477  0.209764
4  0.798935  0.758606 -1.726518
```

正規化

- **正規化**是**將資料的值壓縮在0到1之間**，使得模型更容易找到適當的權重。
- 有大小關係的數據型資料轉成0到1做法是**min max正規化**。
- 類別類型的數據轉成**one-hot encoding**

訓練策略

- **迴歸**為直接預測答案，預測完需要轉換回標準化、正規化前的值。
  - 例如：**預測股市、房價**等

- **分類**需要將答案進行one-hot encoding，如果是二元分類就不用。回歸也可利用範圍的方式轉換成分類，有助於縮小可能的範圍，再進一步預測實際值。
  - 例如：**房價等級** - 高、中、低