

探索性資料分析(EDA)_從資料中選取好的特徵



重要知識點

什麼是好特徵？

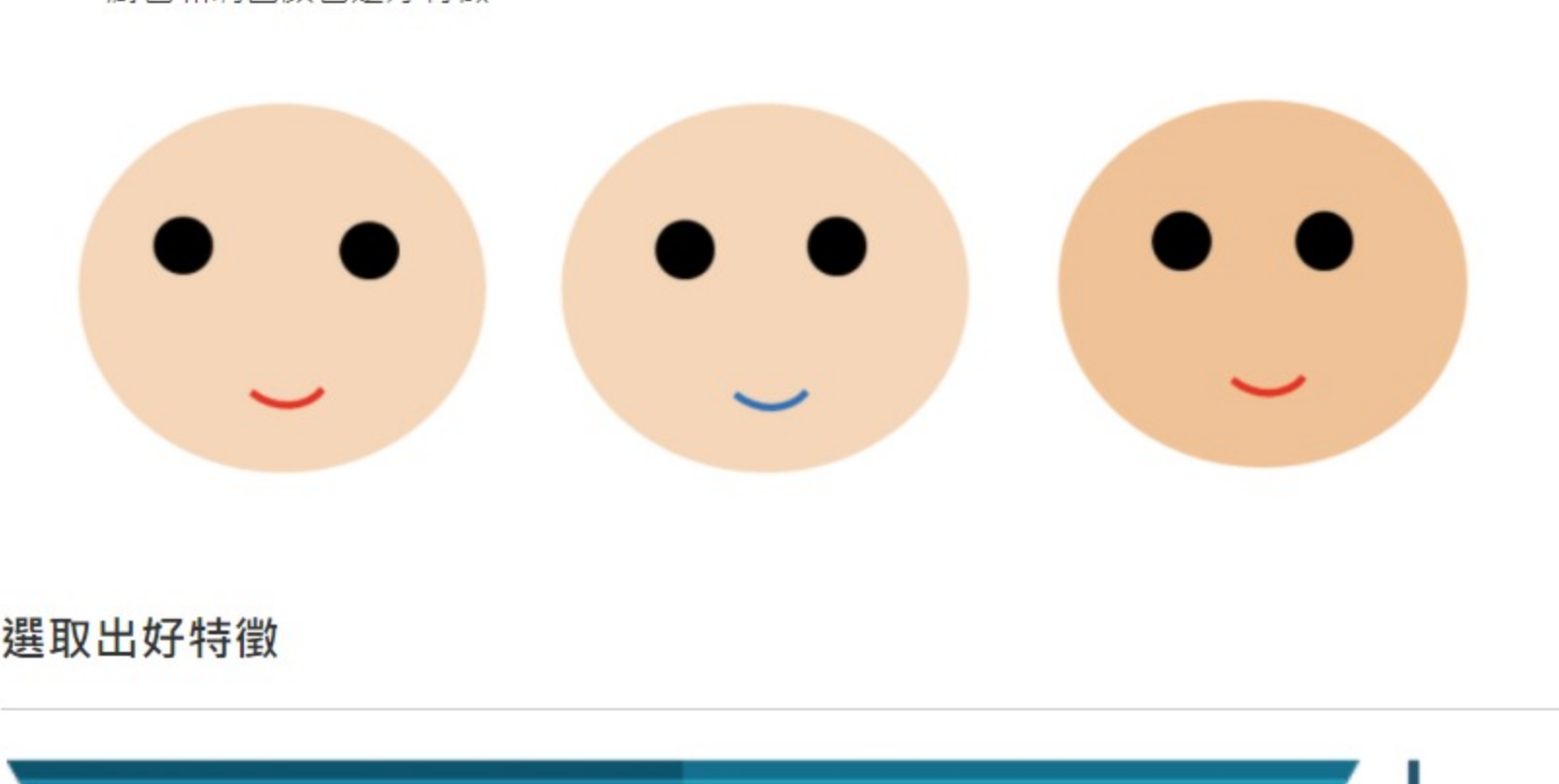
選取出好特徵

好特徵的特性

為什麼要做特徵選擇？



重要知識點



- 掌握好特徵的定義
- 掌握如何挑選出好特徵的方法
- 運用 python 挑選出好特徵

什麼是好特徵？

還記得昨天的情境例子？

- 在原始資料集中有變化性，才能稱為特徵
- 透過這些特徵能把目標做清楚的分類與預測，才能稱為好的特徵
- 膚色和嘴巴顏色是好特徵



選取出好特徵



特徵工程是基於原始資料中萃取出好特徵，藉此改善模型性能的過程。

好特徵的特性



好的特徵，具備以下三種特性：

- 變化性
- 預測性
- 辨識力

為什麼要做特徵選擇？

- 減少特徵數量，使模型泛化能力更強，減少過擬合(overfitting)
- 增強特徵和目標之間的解釋力

課程案例

- 20 筆資料
- 收集到 5 種資料，包含
 - sex：性別
 - insomnia：失眠
 - age：年齡
 - height：身高
 - weight：體重

	sex	insomnia	age	height	weight
0	Male	Y	23	180	100
1	Male	N	40	170	68
2	Male	N	5	100	20
3	Male	N	30	176	70
4	Male	N	1	70	10
5	Female	N	40	160	45
6	Female	Y	16	170	50
7	Female	Y	27	166	58
8	Female	Y	43	155	58
9	Female	N	8	35	17
10	Male	Y	23	170	101
11	Male	N	39	168	65
12	Male	N	5	101	22
13	Male	N	29	175	79
14	Male	N	1	72	12
15	Female	N	42	163	40
16	Female	Y	13	169	53
17	Female	Y	29	163	52
18	Female	Y	41	151	56
19	Female	N	10	40	14

此資料集中，目標資料為失眠這個欄位，我們想找到能挑選出建立失眠模型好的特徵。

特徵選取的三大方法

- 過濾法(Filter)
- 包裝法(Wrapper)
- 嵌入法(Embedded)
- 由於嵌入法是採用建構演算法內的特徵選取方法，必須有演算法的先備知識，所以今天的課程中，以過濾法以及包裝法為主，說明如何透過 python 語法進行特徵選取。
- 運用 sklearn.feature_selection 裡的函數

嵌入法

嵌入法先使用機器學習或模型進行訓練，得到某些特徵的權重係數，根據係數的重要性來選擇特徵，類似過濾法，但是採用訓練的結果來選擇特徵。

過濾法

過濾法是列入一些篩選特徵的標準，把具變化性以及與目標變數相關的特徵，挑選出具變化性以及中高度相關的特徵，方法包含：

- 移除低變異數的特徵
- 單變量特徵選取 (Univariate Feature Selection)
 - 目標變數為離散型，採用卡方檢定(chi2)
 - 目標變數為連續型，可採用 f_regression

1. 移除低變異數的特徵

```
#設定變異數的過濾門檻
filter=VarianceThreshold(threshold=(100))
data_filter= filter.fit_transform(data[num_features]) 特徵
display(data_filter)
print("Variances is %s" % filter1.variances_) #計算每一個特徵的變異數
print("The suport is %s" % filter1.get_support(True)) #確定那些特徵符合條件留下
```

Step1：運用 VarianceThreshold 設定門檻
Step2：透過函數做計算，過濾
Step3：確定哪一些特徵留下來

```
array([[ 23, 180, 100,
        [ 40, 170, 68],
        [ 5, 100, 20],
        [ 30, 176, 70],
        [ 1, 70, 10],
        [ 40, 160, 45],
        [ 16, 170, 50],
        [ 27, 166, 58],
        [ 43, 155, 58],
        [ 8, 35, 17],
        [ 23, 170, 101],
        [ 39, 168, 65],
        [ 5, 101, 22],
        [ 29, 175, 79],
        [ 1, 72, 12],
        [ 42, 163, 40],
        [ 13, 169, 53],
        [ 29, 163, 52],
        [ 41, 151, 56],
        [ 10, 40, 14]], dtype=int64)
```

```
Variances is [ 210.6875 2224.51 715.05 ]
The suport is [0 1 2]
```

代表 x 裡面的第 0,1,2 個位置的特徵會被留下來

觀察：

會發現不一樣的資料變數範圍，有著不同的變異數，所以很難設定統一的過濾標準。
添加標準化的動作後，再建立變異數門檻

下面發現，經過轉換後，變異數大小的順序發生改變囉。

```
Variances is [ 210.6875 2224.51 715.05 ]
The suport is [0 1 2]
```

```
Variances is [0.11943736 0.10580309 0.08634827]
```

Step0：標準化(最大最小值)
Step1：運用 VarianceThreshold 設定門檻
Step2：透過函數做計算，過濾
Step3：確定哪一些特徵留下來

2. 單變量特徵選取

Step1：根據目標變量是連續或離散，來決定判斷的準則。

目標	判斷指標	說明
連續	迴歸係數 (f_regression)	特徵為離散和連續皆可使用
離散 (分類問題)	卡方檢定 (chi2)	目標離散數值型態只有兩類，特徵連續與離散皆可使用，特徵為離散型變數時，使用較佳。 Ex：失眠 vs. 非失眠
	ANOVA檢定 (f_classif)	目標離散數值型態大於多種，特徵連續與離散皆可使用，但為連續型變數時，使用較佳。 Ex：老/中/青

Step2：依照哪一個方法挑選單變量特徵

- SelectKBest：選取 K 個最好的特徵，k 為參數，代表你想選擇多少特徵。
- SelectPercentile：選取多少百分比的特徵，percentile 為參數，代表百分比，用 10 代表 10%。

單變量特徵選取 - 預測失眠狀態(目標變數離散)

Step1：sex 離散型要先轉成數值型態

Step 2：根據目標變量是連續或離散，來決定判斷的準則。
離散型，採用 chi2

Step 3：依照哪一個方法挑選單變量特徵
這邊採用SelectKBest

```
#離散型資料+連續
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
#ValueError: could not convert string to float: 'Male'
#離散型資料要先轉成數值
sex_mapping = {
    'Male': 1,
    'Female': 0}
data['sex1'] = data['sex'].map(sex_mapping)
x=data[['sex1','age','height','weight']]
display(x)
y=data['insomnia']
x_new = SelectKBest(chi2, k=2).fit_transform(x, y)
x_new.shape
display(x_new)
```

以卡方分配來看，和失眠狀態最相關的變數為 height weight

單變量特徵選取 - 目標體重(目標變數連續)

Step 1：sex 與 insomnia 離散型要先轉成數值型態

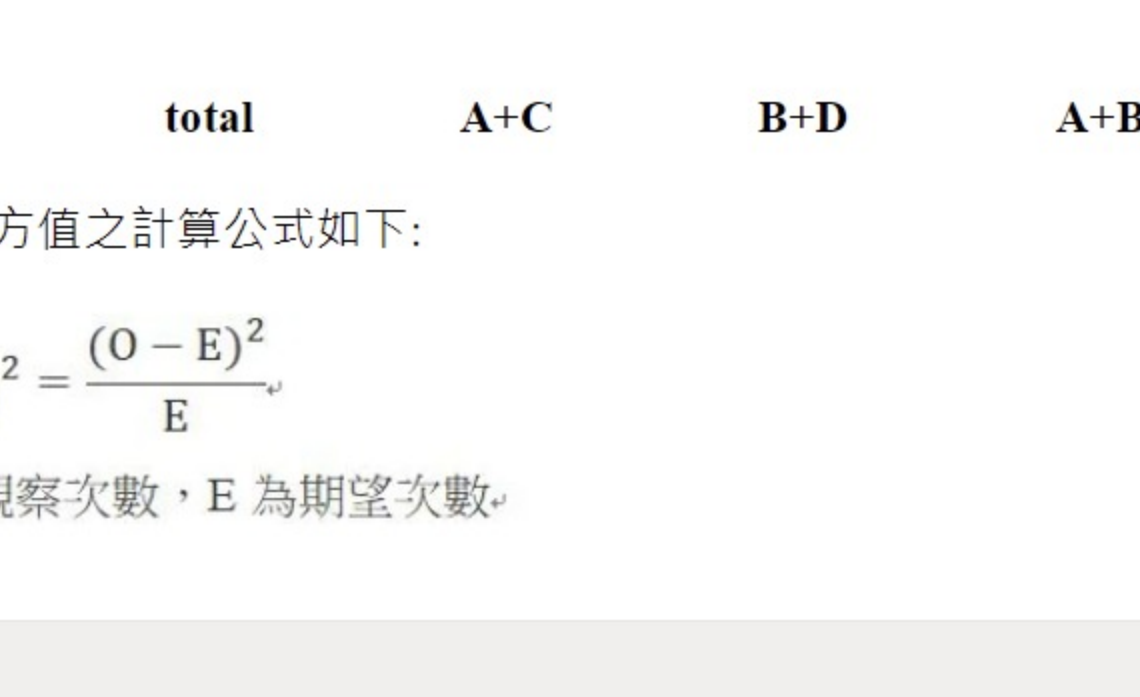
Step 2：根據目標變量是連續或離散，來決定判斷的準則
連續型，採用 f_regression

Step 3：依照哪一個方法挑選單變量特徵
這邊採用 SelectPercentile

```
## 選取
# 依照失眠狀態、身高預測 體重
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_regression
#ValueError: could not convert string to float: 'Male'
#離散型資料要先轉成數值
insomnia_mapping = {
    'Y': 1,
    'N': 0}
data['insom1'] = data['insomnia'].map(insomnia_mapping)
sex_mapping = {
    'Male': 1,
    'Female': 0}
data['sex1'] = data['sex'].map(sex_mapping)
x=data[['sex1','age','height','insom1']]
display(x)
y=data['weight']
x_new = SelectPercentile(chi2, percentile=50).fit_transform(x, y)
x_new.shape
display(x_new)
```

包裝法

- 包裝法將特徵選擇看作是搜索問題，根據某一種評量標準，每次選擇某些特徵或排除某些特徵，常用的方法為遞歸特徵消除(RFE)。
- 遞歸特徵消除(RFE)
- 根據你的問題是離散或連續，選擇帶有 "coef_" 和 "feature_importances_" 的模型
- 例如：
 - SVG(kernel="linear")
 - LogisticRegression
 - 關於上述模型的意義，後續高階課程會在詳細講述。



資料來源：Feature Selection Methods for Data Science

運用 python 執行包裝法

Step 1：sex 離散型要先轉成數值型態

Step 2：根據目標變量是連續或離散，來決定判斷的準則。
離散型，SVC(kernel="linear")

Step 3：設定 RFE 裡面的參數

- n_features_to_select：最後要選擇留下多少特徵。
- Step：刪除法，每一步刪除多少特徵。

Step 4：fit(x,y)：每一步都依不同的特徵組合建立模型，判斷最終要選擇那些特徵

Step 5：透過 support_ 呈現包裝法搭配 SVC 下，選擇最好的特徵，用 True 來表示

Step 6：透過 ranking_ 呈現每個特徵對於模型的重要性，1 代表被選中的特徵，2 代表次要的特徵，依此類推

```
from sklearn.datasets import make_friedman1
from sklearn.feature_selection import RFE
from sklearn.svm import SVC
#ValueError: could not convert string to float: 'Male'
#離散型資料要先轉成數值
sex_mapping = {
    'Male': 1,
    'Female': 0}
data['sex1'] = data['sex'].map(sex_mapping)
x=data[['sex1','age','height','weight']]
display(x)
#insomnia_mapping = {
#    'Y': 1,
#    'N': 0}
#data['insom1'] = data['insomnia'].map(insomnia_mapping)
y=data['insom1']
estimator = SVC(kernel="linear")
selector = RFE(estimator, n_features_to_select=2, step=1)
selector = selector.fit(x, y)
```

```
## True= selected feature
print(selector.support_) [ True False True]
#Selected (i.e., estimated best) features are assigned rank 1.
ranking=selector.ranking_ [1 2 3 1]
print(ranking)
rfe_feature = x.loc[:,selector.support_].columns.tolist()
print(rfe_feature) [ 'sex1', 'weight' ]
```

延伸閱讀

什麼是卡方檢定？

網站：卡方檢定-獨立性檢定(The Chi-Squared Test of Independence)-統計說明與SPSS操作

卡方檢定用於分析兩類別變數間的關係，在今天的課程範例中，卡方檢定也應用於一個類別型與連續型資料，主要因為函數把連續型資料，連續型資料，對右表格而言為稀疏資料，在函數中對於稀疏資料有進行處理，所以能做卡方檢定的計算。

對卡方檢定有興趣的學員，可以詳細閱讀延伸閱讀。

		Disease(b)		total
		yes	no	
Exposure(a)	yes	A	B	A+B
	no	C	D	C+D
total		A+C	B+D	A+B+C+D

(一)卡方值之計算公式如下：

$$\sum x_{ij}^2 = \frac{(O - E)^2}{E}$$

O 為觀察次數，E 為期望次數

下一步：閱讀範例與完成作業