資料科學(大數據):

* 抓取
* 分析:分為兩大類
* 機器學習(傳統統計):處理結構(表格)類型的
* 深度學習(現代設計、神經網路):處理抽象類型的(ex:語言、音樂、圖)

環境:

Google corlab

老師範例: <https://github.com/Elwing-Chou/ml0930/>

輸出:

* 類別(選擇題、分類(classification))
* 數值(計算題、迴歸(regression) )
* 無答案的選擇題(靠input相似度猜答案 分群(cluster))

表格:pandas

如果pandas爆掉就要使用 modin(<https://github.com/modin-project/modin>)

Gini:

猜錯的機率是多少，係數要越低越好。

Entropy:

過擬和:訓練太好導致反正錯誤率提高。

處理方法:

* 前剪枝:設最大深度(max\_depth)。
* 後剪枝:sk-learn不支持

\*可解釋性

MSE(mean sguaired error):分布的廣度



RMSE:開根號MSE

MAE=1/NE|pre

r^2 score

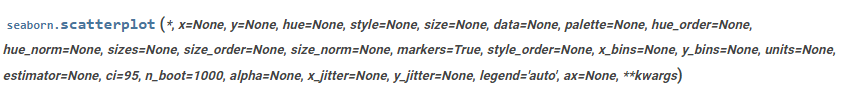
kmeans非常重要

分群

* 已知:選
* 未知:try k試過一遍

Silhouette score: 0最差 1最好

畫圖最基礎:mat 比較好用 seaborn



Hue是加上顏色

殘體字轉繁體字:opencc

中文分詞:jieba

分詞時:辭典遇到

* 混淆:都是看上下文
* 不認識

jieba.set\_dictionary(“要設定的字典”)

jieba.load\_userdict("自己訂的字典")

關鍵詞(重點)

1. 次:多,重要提升
2. 常用度:常用,重要下降

重要性=次數\*(1/常用度)

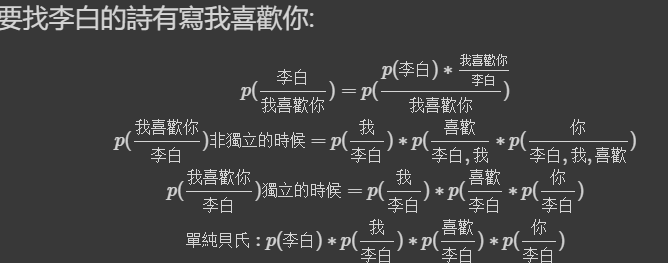
常用度=詞(文章)/文章數

維度災難:欄位太多

p(A)−>A發生機率

p(A,B),p(A⋃B)−>A,B同時發生的機率

p(A|B),p(ab)−>B條件下,A發生的機率



有時文字使用獨立或非獨立都有東西求不出來，所以直接使用單純貝氏就好。

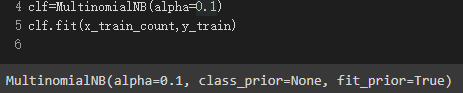
**\*\*\*\*\*apply**

**Train:fit+transform**

**Test:不用fit 只用transform**

**如果test裡面出現Train沒有的東西直接視而不見。**

**假設判斷一封信裡有出現你就不是垃圾信，如果有封信是寫\*\*\*\*\*\*你，應判斷為垃圾，但是因有你這個字變成非垃圾，解決方法就是使用Smoothing。**

**\*\*\*\*\*Smoothing:避開0**

**因為唐詩的字可能都出現一次兩次，所以alpha+1太大會出現錯誤，所以把alpha降低可提高命中率。**