IAI大數據競賽訓練

01-檔案讀取與儲存

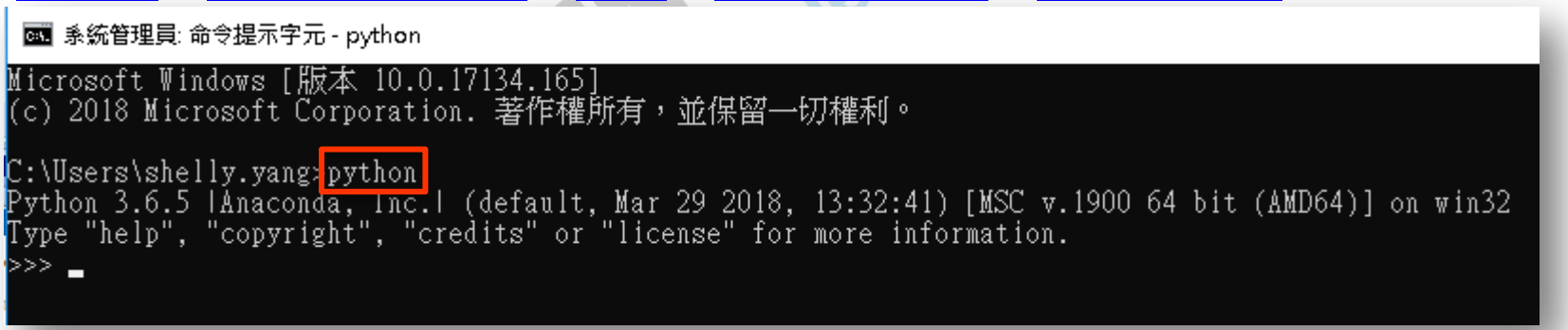
講師：楊甯雅

Agenda

- 安裝Python外部套件
- 自動生成讀檔路徑與讀檔
- 活用DataFrame格式
- 分析結果存檔

安裝Python外部套件

1. 設定環境變數(以anaconda為例)
   ① anaconda中pip.exe路徑
   C:\Users\shelly.yang\AppData\Local\Continuum\anaconda3\Scripts
   ② python.exe路徑
   C:\Users\shelly.yang\AppData\Local\Continuum\anaconda3
   ③ 將上述路徑加入環境變數中
   控制台➔所有控制台項目➔系統➔進階系統設定➔環境變數(N)➔PATH



安裝Python外部套件

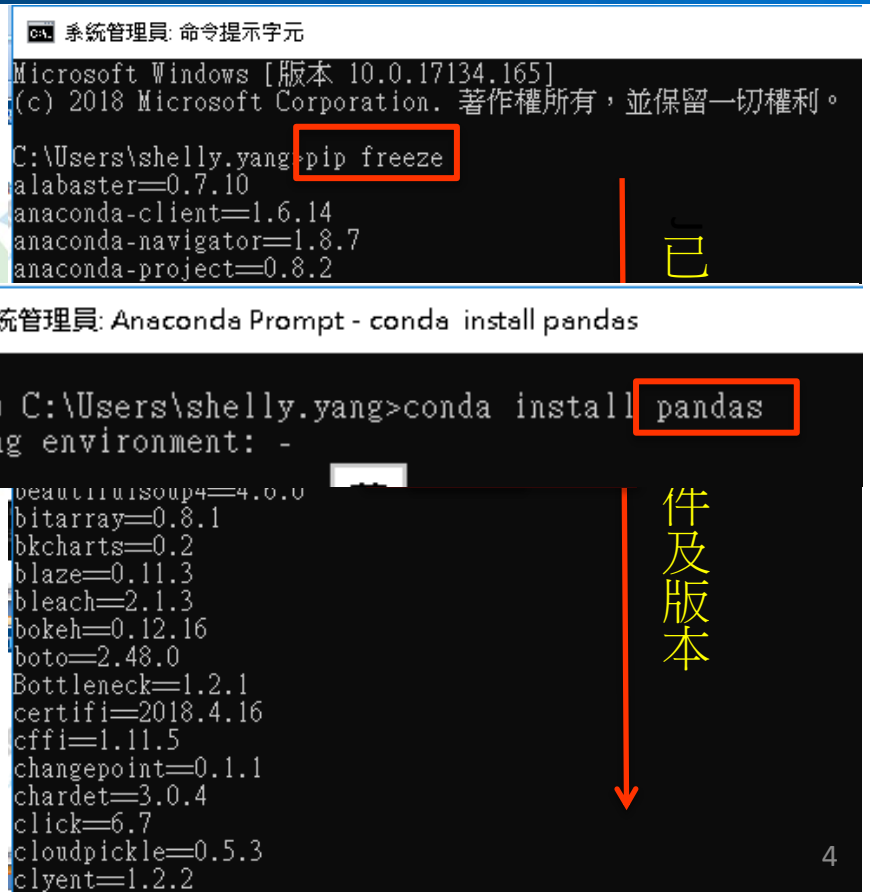
2. 查詢已安裝套件

3. 安裝外部套件

   1) 在console下指令安裝

   2) 下載set.py或.whl檔安裝

   3) 在Anaconda Prompt下指令安裝



自動生成讀檔路徑與讀檔

競賽數據

Training Data (6 Gb)

共2把刀具，每把含315次的銑切，每次銑切結束後會蒐集一筆磨損值與銑切期間的高頻數據
(各約20萬筆振動、力量與音洩訊號特徵)

Testing Data (3 Gb)

1把刀具，含315次的銑切作動的高頻數據

Train_A
- Train_A_001
- Train_A_002
- ⋮
- Train_A_314
- Train_A_315

Train_A_wear

cut	flute_1	flute_2	flute_3
1	31.41636	19.48369	21.74806
2	34.89277	23.47305	24.92596
⋮			
314	195.0154	209.867	201.2441
315	196.1648	210.9194	202.1496

Train_B
- Train_B_001
- Train_B_002
- ⋮
- Train_B_314
- Train_B_315

Train_B_wear

cut	flute_1	flute_2	flute_3
1	32.31711	48.89262	37.72083
2	37.91488	49.57082	37.72083
⋮			
314	172.2646	163.9247	157.5121
315	172.6868	164.6379	158.1925

Test
- Test_001
- Test_002
- ⋮
- Test_314
- Test_315

額外關於資料獲取的資訊：
資料收集自一把六毫米碳化鎢三刃球型立銑刀
主軸轉速是10400 RPM
料件進給速度是一分鐘1555毫米(mm/min)
Y方向切削 深度是0.125毫米(mm)
Z方向切削 深度是0.2毫米(mm)
資料獲取頻率是50 KHz

自動生成讀檔路徑與讀檔

用程式生成符合自己目的的檔案路徑：套件**glob**

glob.glob(pathname)

查找路徑

於D:/Project/IAI/路徑下搜尋檔案

```
glob.glob('D:/Project/IAI/*')
glob.glob('D:/Project/IAI/*.csv')
glob.glob('D:/Project/IAI/*.jpg')
.....
etc.
```

```
['D:/Project/IAI/raw data/Train_A/Train_A\\Train_A_001.csv',
 'D:/Project/IAI/raw data/Train_A/Train_A\\Train_A_002.csv',
 'D:/Project/IAI/raw data/Train_A/Train_A\\Train_A_003.csv',
 'D:/Project/IAI/raw data/Train_A/Train_A\\Train_A_004.csv',
                                .
                                .
                                .
 'D:/Project/IAI/raw data/Train_A/Train_A\\Train_A_311.csv',
 'D:/Project/IAI/raw data/Train_A/Train_A\\Train_A_312.csv',
 'D:/Project/IAI/raw data/Train_A/Train_A\\Train_A_313.csv',
 'D:/Project/IAI/raw data/Train_A/Train_A\\Train_A_314.csv',
 'D:/Project/IAI/raw data/Train_A/Train_A\\Train_A_315.csv']
```

自動生成讀檔路徑與讀檔

套件**pandas**：主要提供Panel、DataFrame、Series三種資料結構

檔案格式	檔案讀取	寫出儲存
CSV	pandas.read_csv	to_csv
JSON	pandas.read_json	to_json
HTML	pandas.read_html	to_html
EXCEL	pandas.read_excel	to_excel
SQL	pandas.read_sql	to_sql

PS: csv檔的檔案讀取路徑需全為英文

自動生成讀檔路徑與讀檔

練習:

```
import pandas as pd
import glob

trainSet = ['Train_A', 'Train_B']

for Set in trainSet:

    path = 'D:/Project/IAI/raw data/'+Set+'/'
    trainY = pd.read_csv(path+Set+'_wear.csv')

    allFolder = glob.glob(path+Set+'/*.csv')
    for (cut, folder) in enumerate(allFolder):
```

import 套件

Set依序為trainSet中的字串

folder位置

folder依序為allFolder字串

DataFrame

pandas讀入檔案後，資料結構為data frame

1. 運用data frame運算，降低執行時間
2. 便於活用於檔案/表格合併

查看目前資訊

```
#回傳列數與欄位數
df.shape
#計算所有欄位之描述性統計量
df.describe()
#回傳前三筆觀測值
df.head(3)
#回傳最後三筆觀測值
df.tail(3)
#欄位名稱
df.columns
```

處理空值

```
#將有遺失值的觀測值刪除
df.dropna()
#將空值補為0
df.fillna(0)
```

合併檔案(依據欄位)

```
pd.merge(A,B,left_on=[欄位(A)],right_on=[欄位(B)])
```

依據不同類別統計個數

```
df.groupby(["欄位名稱1","欄位名稱2","欄位名稱3",...])
```

https://blog.csdn.net/wr339988/article/details/65446138

DataFrame

```
def featureTD(DF):

    colnames = DF.columns
    factorName = []
    rslt = []
    for (colIDX,cols) in enumerate(colnames):

        df = DF.ix[:,colIDX]

        # maximum
        factorName.append(cols+'_Max')
        rslt.append(df.max())

        # mean
        factorName.append(cols+'_Mean')
        rslt.append(df.mean())

        # Root Mean Square
        factorName.append(cols+'_RMS')
        rslt.append(np.sqrt((df**2).mean()))

        # Standard Deviation
        factorName.append(cols+'_Std')
        rslt.append(df.std())
```

```
        # Skewness
        factorName.append(cols+'_Skewness')
        rslt.append(df.skew())

        # Kurtosis
        factorName.append(cols+'_Kurtosis')
        rslt.append(df.kurtosis())

        # Peak to Peak
        factorName.append(cols+'_P2P')
        rslt.append(df.max()-df.min())

        # Crest Factor
        factorName.append(cols+'_CF')

    rslt.append(df.max()/np.sqrt((df**2).mean()))

    return factorName, rslt
```

分析結果存檔

分析結果結構為data frame

```
df = pd.DataFrame({'min':minimum, 'max':maximum,  'average':average},
                  index = ['Bag','AdaBoost','Kpca_Bag','Kpca_AdaBoost'])

writer = pd.ExcelWriter(path+ criteria+'.xls')

df.to_excel(writer,criteria)
writer.save()
```

藉由dictionary格式將結果轉為data frame

Sheet名稱

存檔路徑+檔名+'.xls'

www.innolux.com

INNOLUX
群創光電