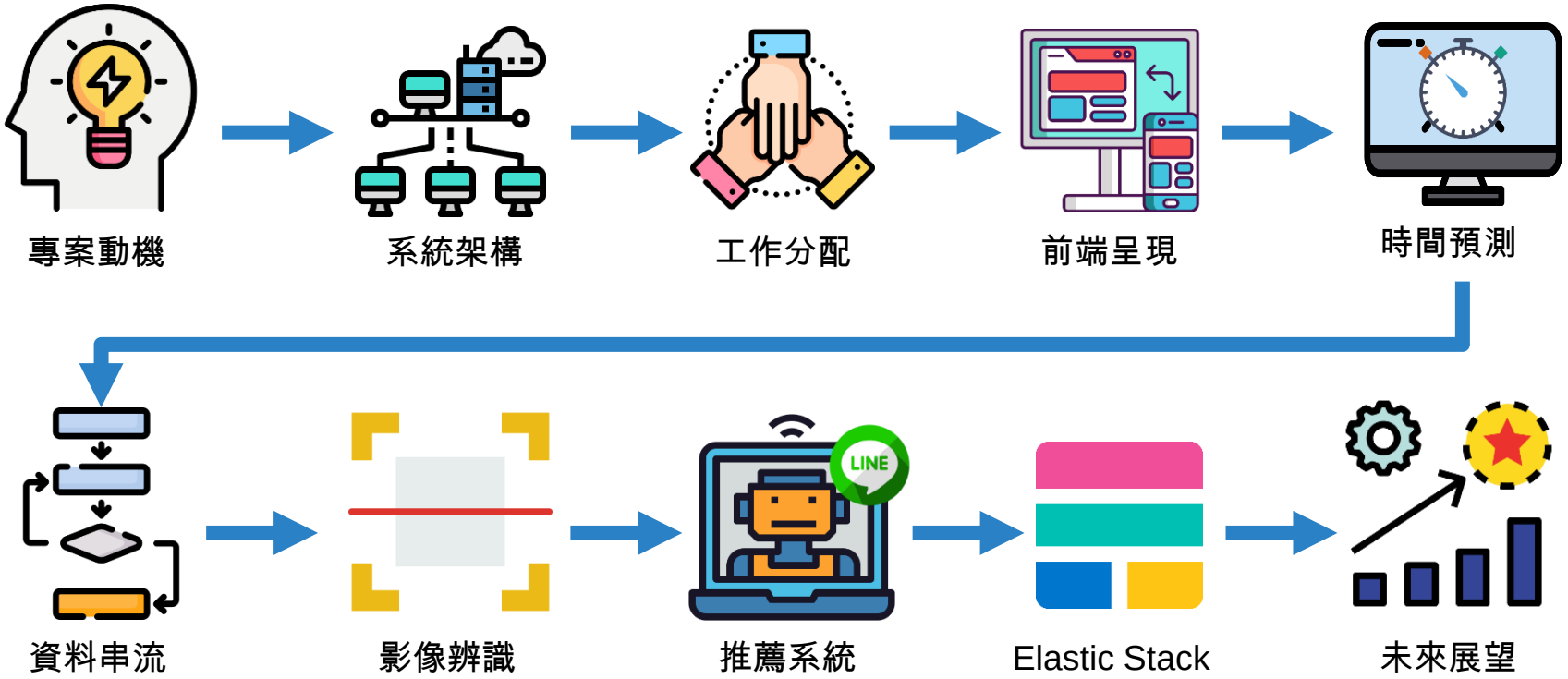
Hospital Assistant

Hospital Assistant

醫療助手

報告流程


專案動機
系統架構
工作分配
前端呈現
時間預測
資料串流
影像辨識
LINE
推薦系統
Elastic Stack
未來展望

專案動機

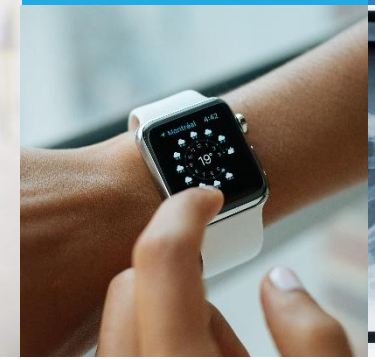
報告人：陳亞生


MEDICAL
Health Care
Doctor
Hospital
Pharmacist
Nurse
Dentist
First Aid
Surgeon
Emergency
MEDICAL
MEDICAL
MEDICAL

專案動機?

時間預測

預測使用精密儀器
做檢查的時間



影像辨識

初步辨識並挑選出，
過濾無病變影像

推薦系統

提供檢查資訊及推
薦醫療文章資訊

報告人：陳亞生

時間預測

Why?

預約**時間排程太長**

Target

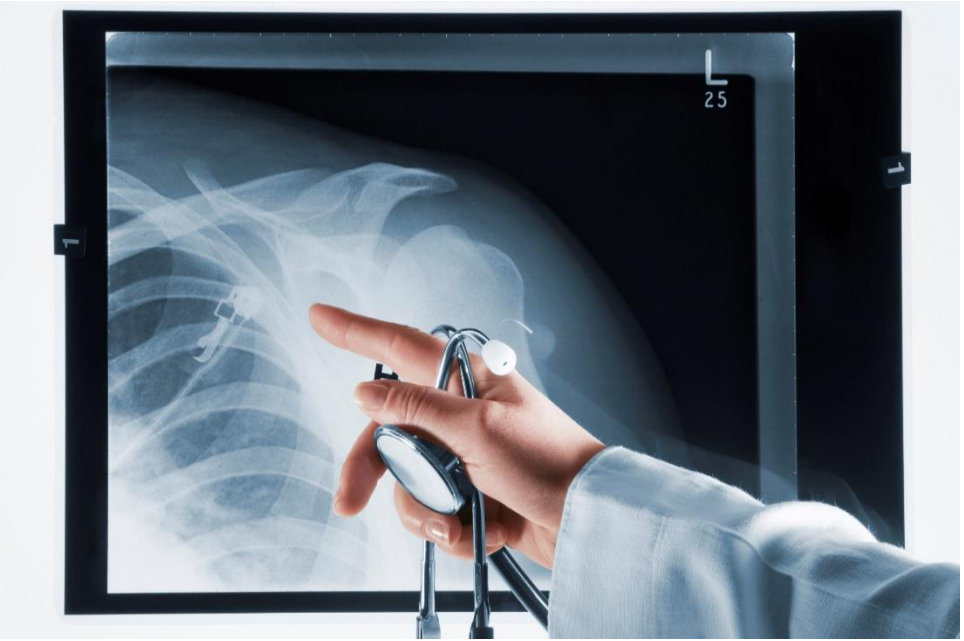
透過AI模型，去**預測**更**精準**的**預估時間**

Result

縮短排程時間，**增加醫院收益**

報告人：陳亞生

影像辨識



Why?

影像結果的初步評估，**數量**造成醫生**很大的負擔**

Target

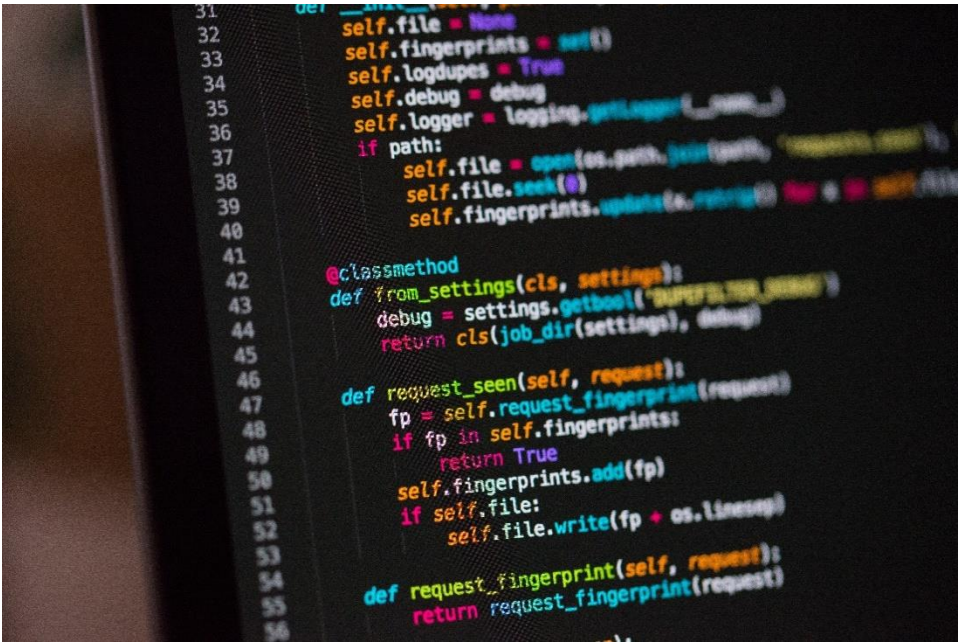
透過**影像辨識**將**沒有病變**的影像先**挑出來**

Result

減少醫生**負擔**，**減少人力**，進而**提升**院內**業務績效**

報告人：陳亞生

推薦系統



Why?

根據病人情況推薦文章

Target

透過演算法推薦更貼近使用者的文章

Result

利用使用者資訊提供合適的營運配置，以符合當時使用者關注疾病的門診需求

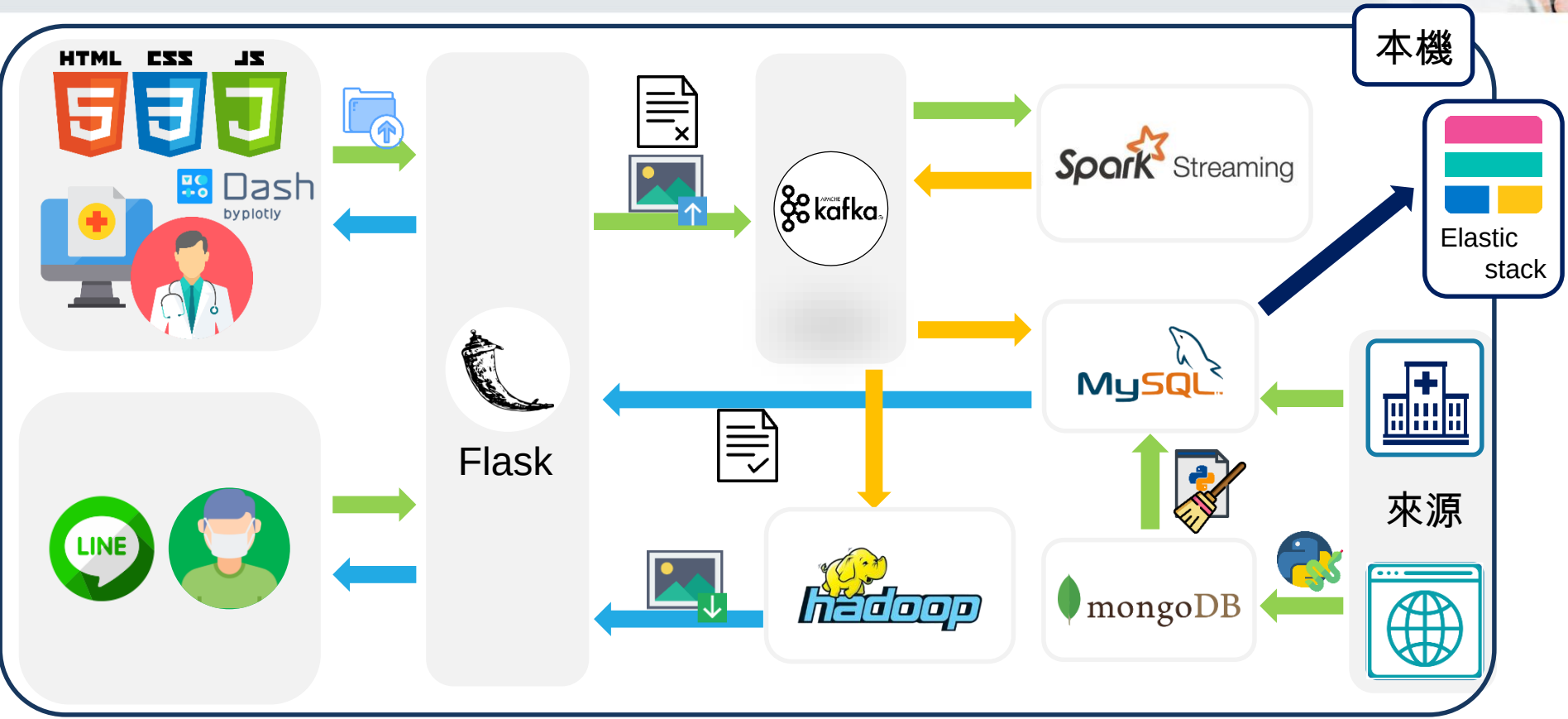
報告人：陳亞生

系統架構

報告人：陳亞生


MEDICAL
Health Care
Doctor
Hospital
Pharmacist
Nurse
Dentist
First Aid
Surgeon
Emergency
MEDICAL
MEDICAL
MEDICAL

系統架構


本機
HTML
CSS
JS
Dash
by plotly
LINE
Flask
kafka
Spark Streaming
Elastic stack
MySQL
來源
hadoop
mongoDB

工作分配

報告人：陳亞生


MEDICAL
Health Care
Doctor
Hospital
Pharmacist
Nurse
Dentist
First Aid
Surgeon
Emergency
MEDICAL
MEDICAL
MEDICAL

工作分配


組長


副組長

Hospital Assistant	陳亞生	盧冠宏	陳泓睿	賀俊賢	鄧鈺蓉	陳彥蓉	李建德	郭勇宗
前端呈現	✚		✚					
時間預測	✚		✚		✚	✚		
影像辨識		✚						✚
推薦系統					✚		✚	
Line-Bot					✚	✚	✚	
系統架構	✚	✚		✚				
Elastic stack				✚				

報告人：陳亞生

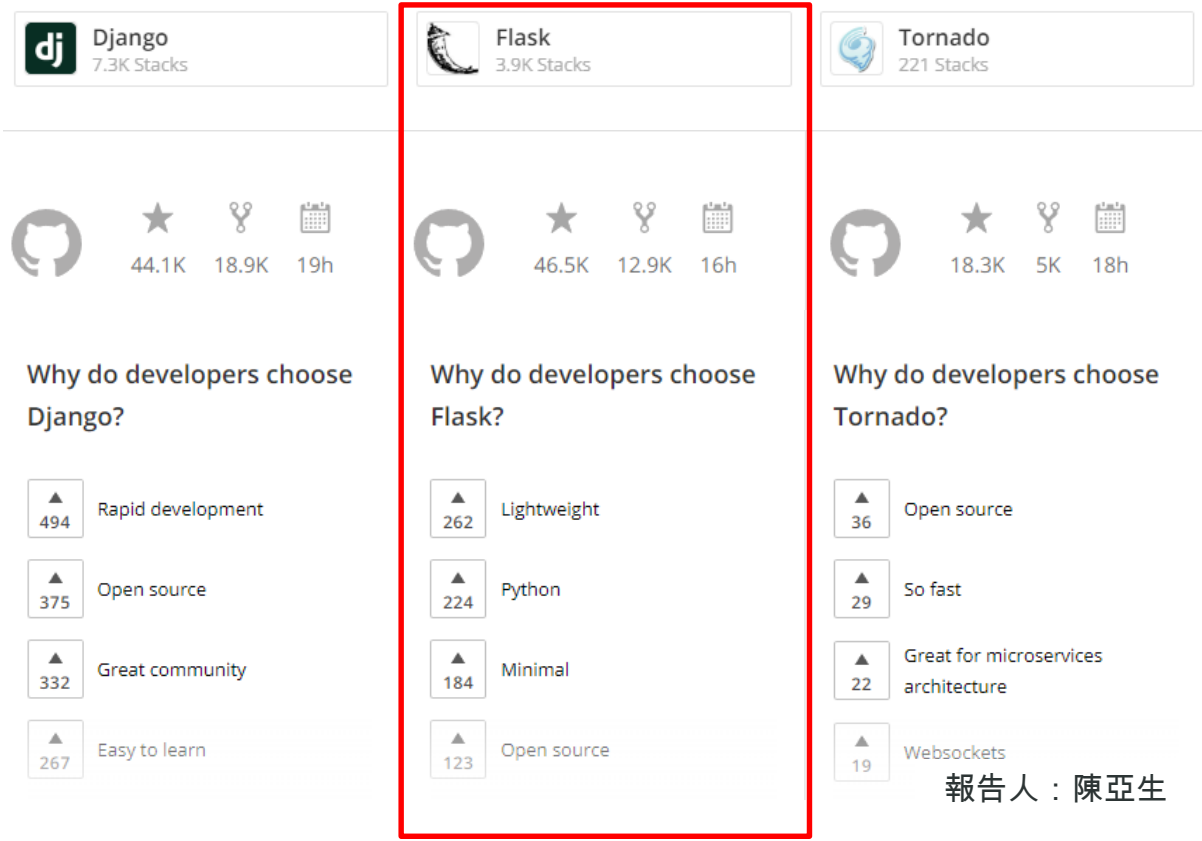
前端呈現

報告人：陳亞生


MEDICAL
Health Care
Doctor
Hospital
Pharmacist
Nurse
Dentist
First Aid
Surgeon
Emergency
MEDICAL
MEDICAL
MEDICAL

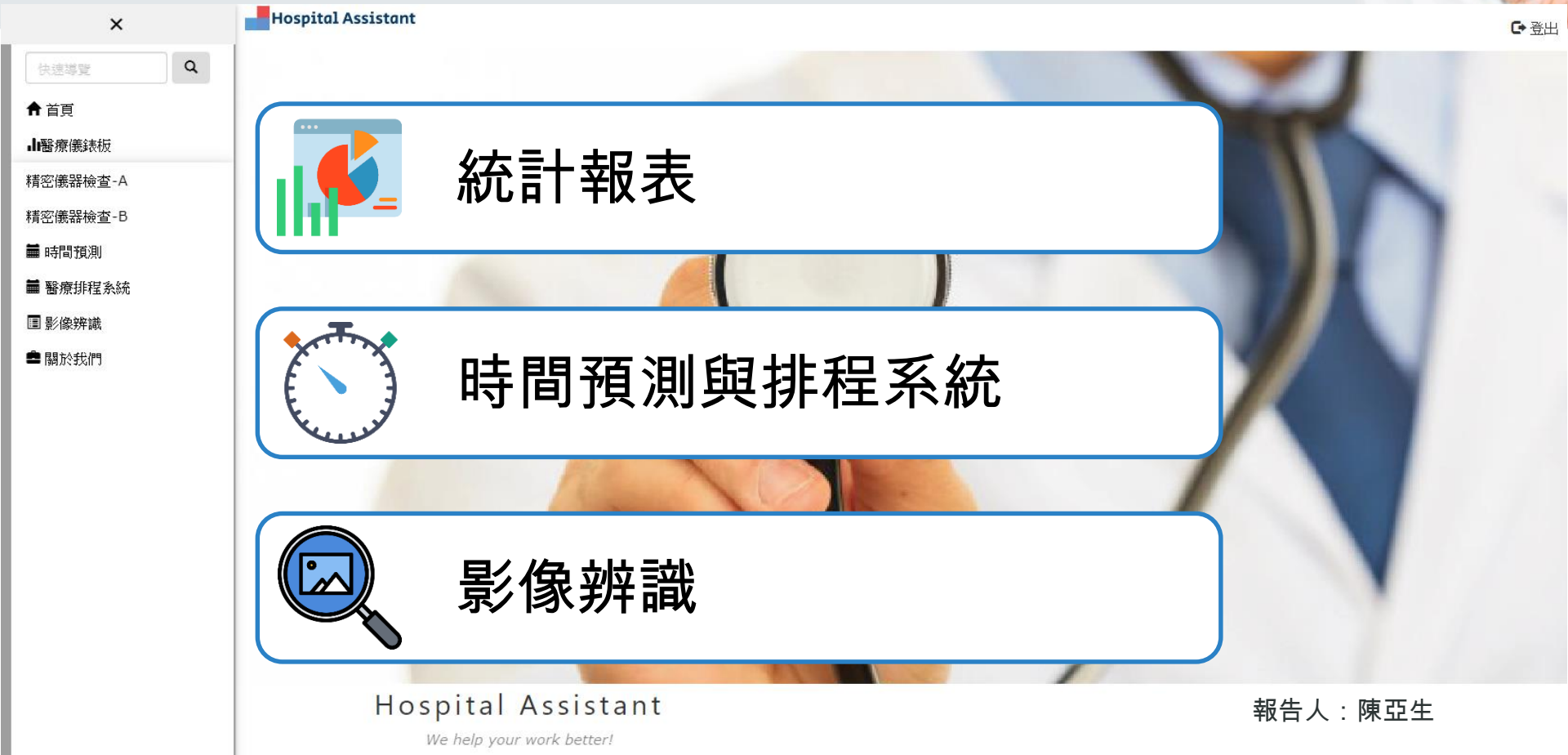
WHY Flask?

1. Flask 是一個輕量級網站框架
2. 給予開發者很大彈性
3. 與Python串接API



報告人：陳亞生

Web功能介紹



報告人：陳亞生

統計報表

Hospital Assistant

登出

精密儀器檢查-A

報告人：陳亞生

請選擇年份與月份

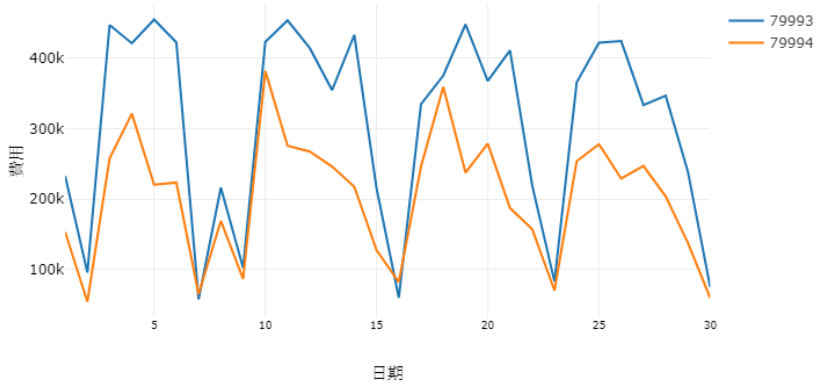
2017 2018 2019

June

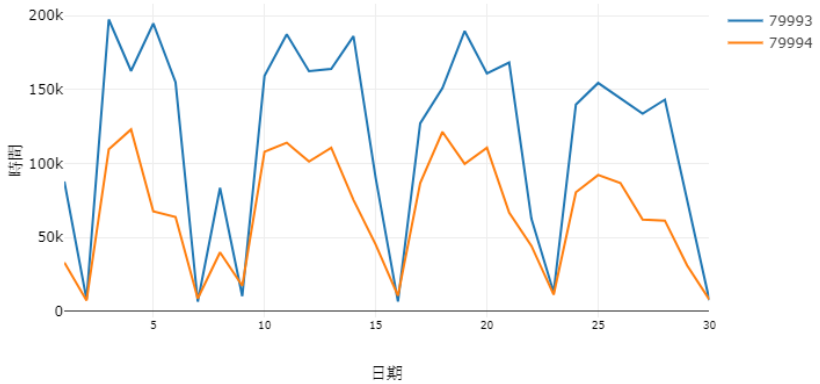
檢查項目統計表

79994 × ▾ 79993 × ▾

每日費用



每日時間

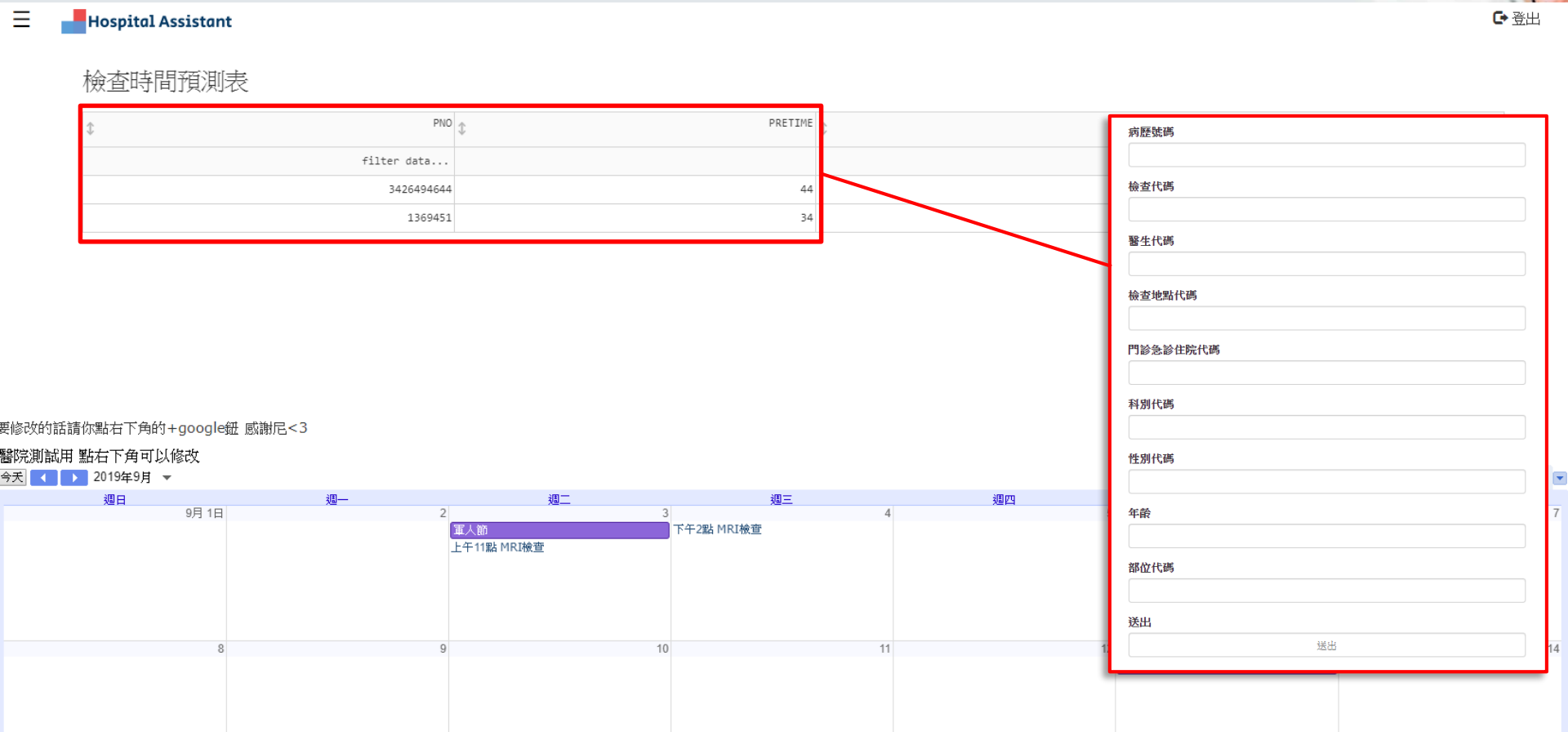


快速導覽
首頁
關於我們

Hospital Assistant

We help your work better!

時間預測與排程 Web & Linebot

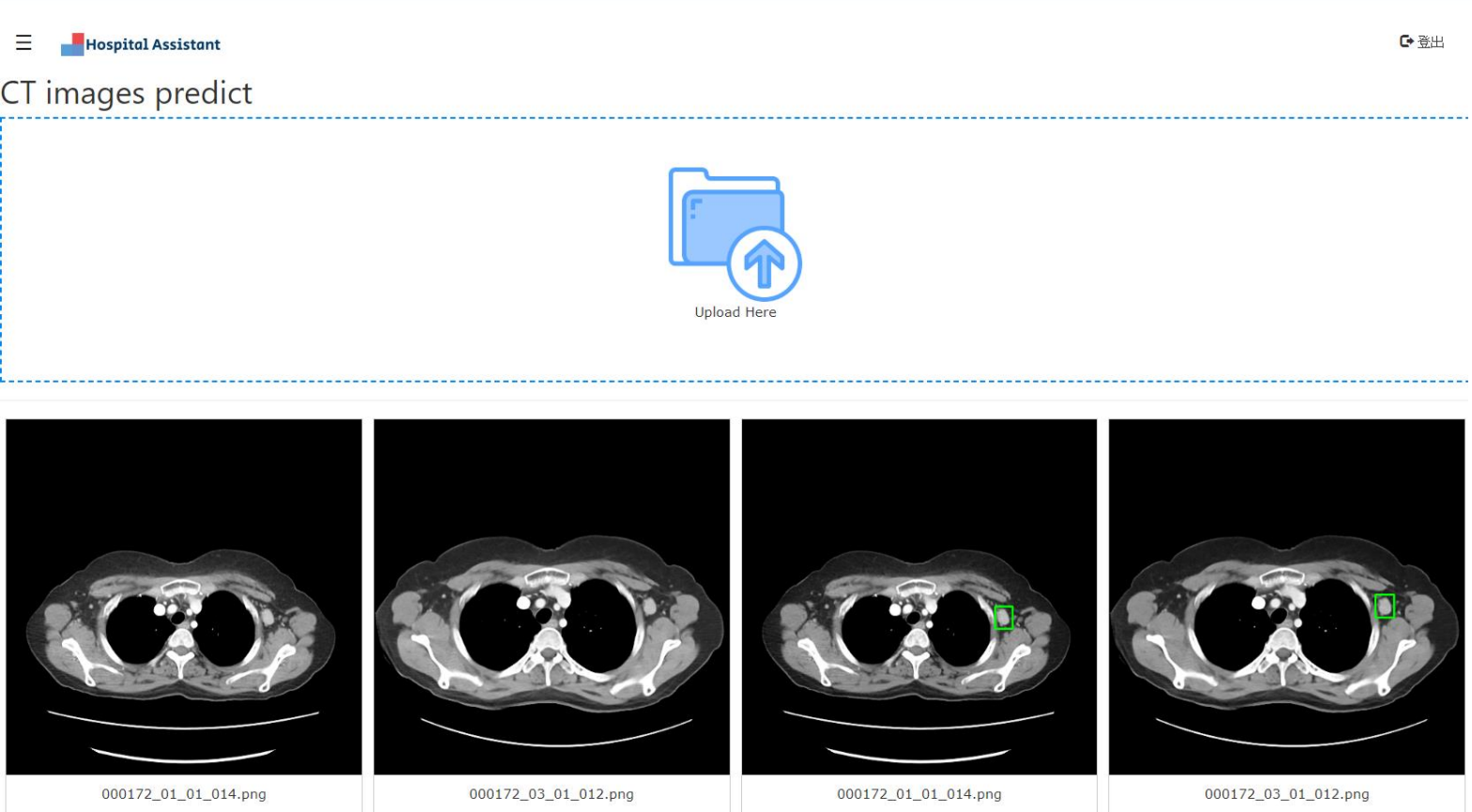


前端展示

新增病患資料

- Web & Line

影像辨識


Hospital Assistant
登出
CT images predict
Upload Here
000172_01_01_014.png
000172_03_01_012.png
000172_01_01_014.png
000172_03_01_012.png

報告人：陳亞生

登出

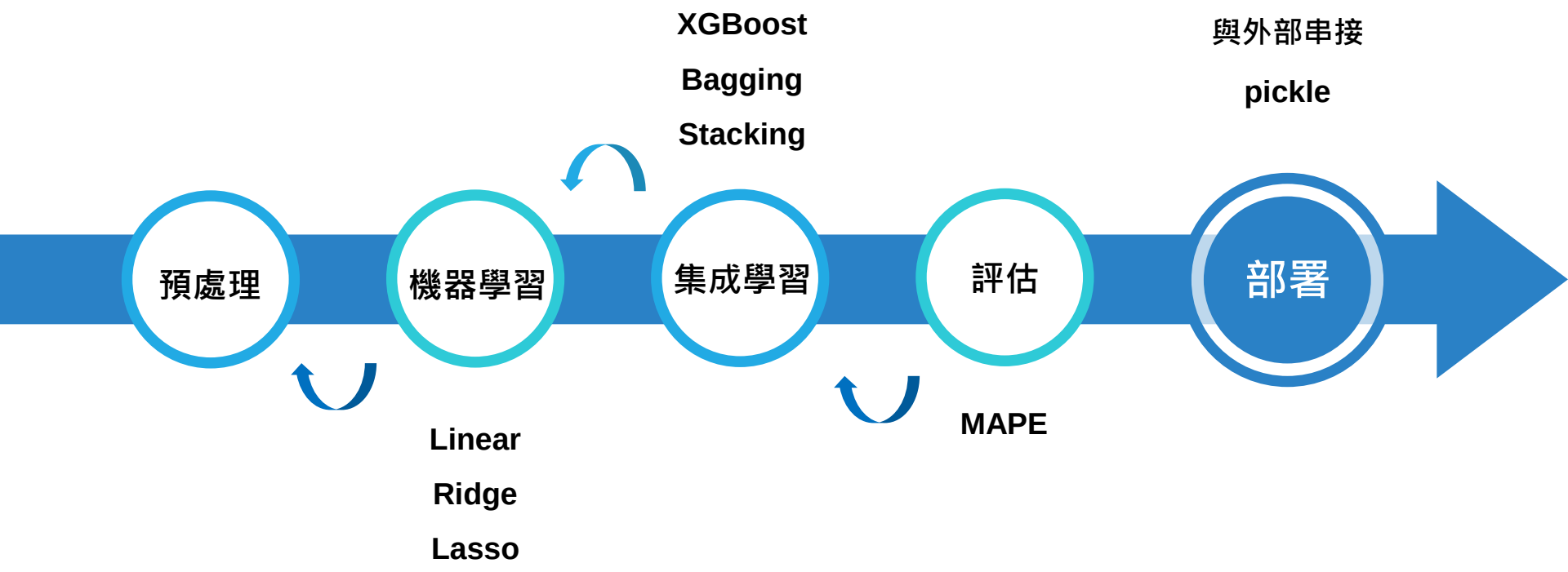
Upload Here

時間預測

時間就像海綿裡的水，只要願擠，總還是有的。——魯迅

報告人：陳彥蓉

預測流程


XGBoost
Bagging
Stacking
與外部串接
pickle
預處理
機器學習
集成學習
評估
部署
Linear
Ridge
Lasso
MAPE

報告人：陳彥蓉

資料的預處理

```
#標準化
min = data['AGE'].min()
max = data['AGE'].max()
data['AGE'] = (data['AGE'] - min)/(max-min)
print(data['AGE'])
```

挑選欄位

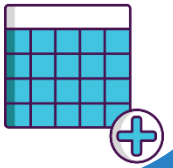
重複資訊
多餘資訊
資料格式

資料清理

數值型變數處理

Z-score
Min-Max

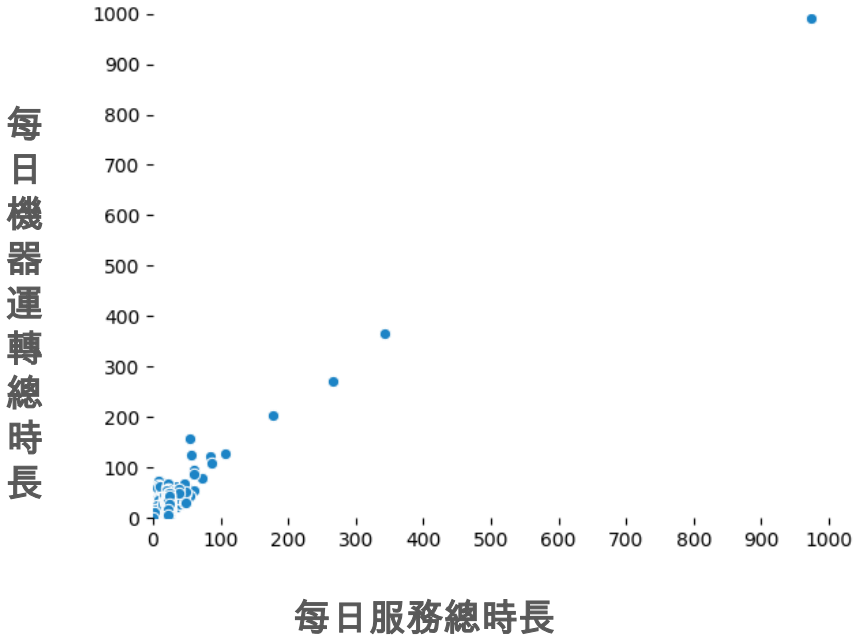
類別型變數處理



報告人：陳彥蓉

資料處理的重要性 未處理的資料

轉換時間格式 ➡ 計算時間差 ➡ 資料分組與聚合

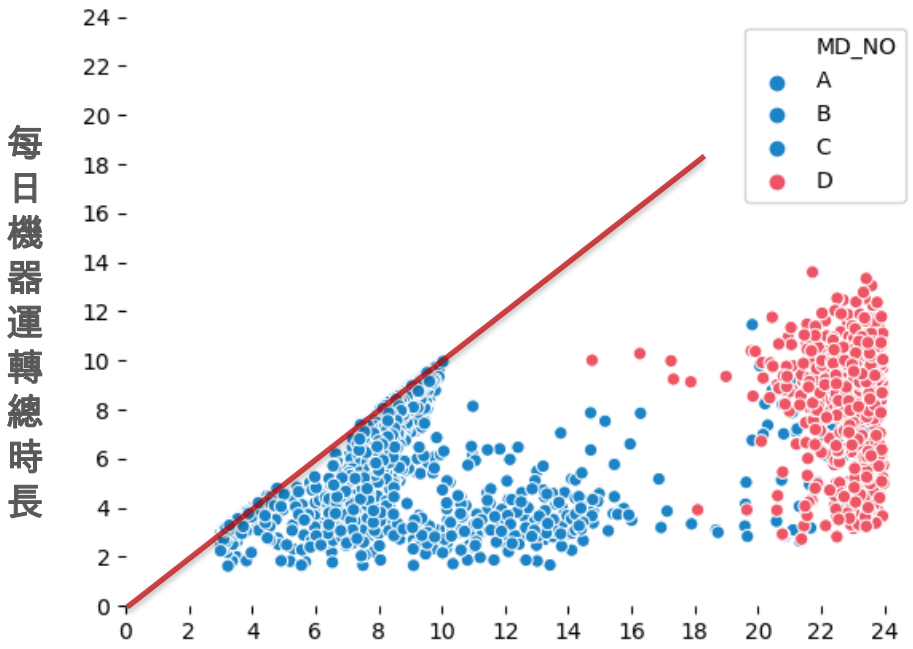


- ❌ 服務時間不合理
- ❌ 運轉時間不合理
- ❌ 運轉大於服務時間

報告人：陳彥蓉

資料處理的重要性 處理過的資料

機器每日使用效率



- ☑ 交錯時間
- ☑ 輸入錯誤的時間
- ☑ 排除時間過長的資料
- ☑ 空值

報告人：陳彥蓉

類別型變數方式

標籤化	增加維度	敘述統計
Label Encoding	Dummies	Mean Encoding
	One Hot Encoding	中位數
		排名

機台代碼
A
B
C

→

機台代碼
1
2
3

→

機台A	機台B	機台C
1	0	0
0	1	0
0	0	1

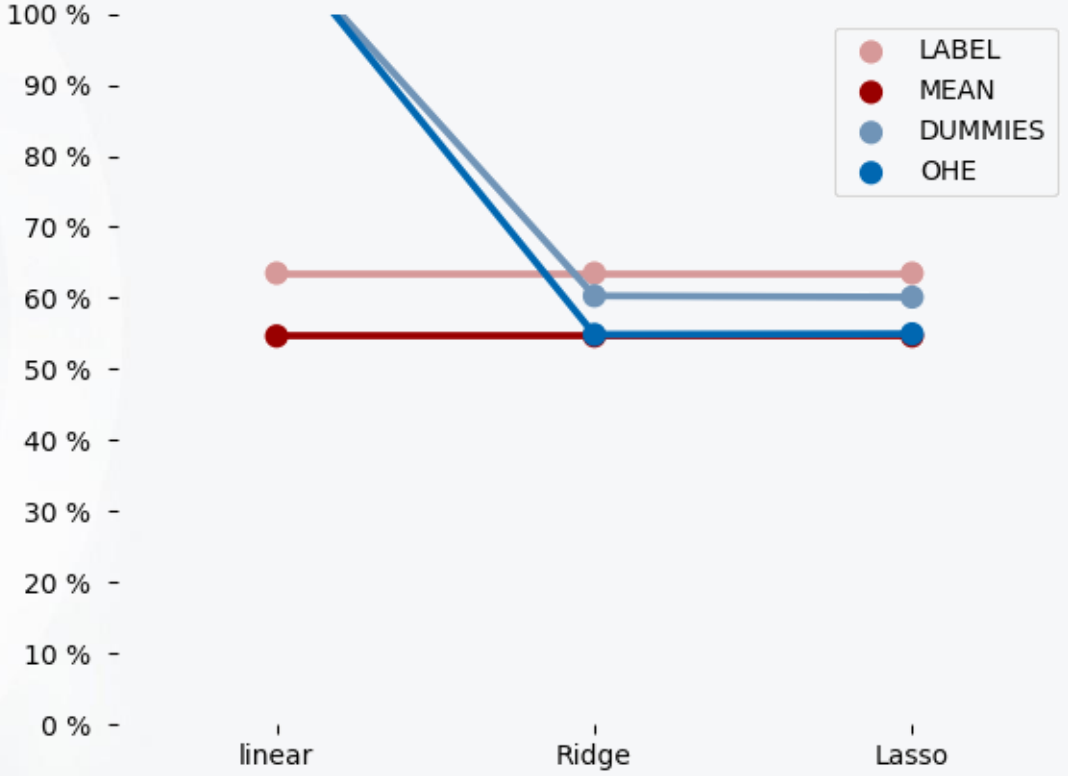
→

機台代碼	平均每人使用時間
A	9分鐘
B	14分鐘
C	15分鐘

報告人：陳彥蓉

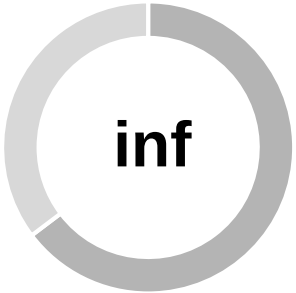
轉換方式比較

MAPE

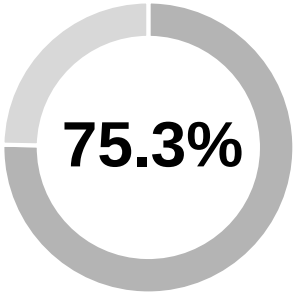


報告人：陳彥蓉

類別型變數處理的MAPE比較

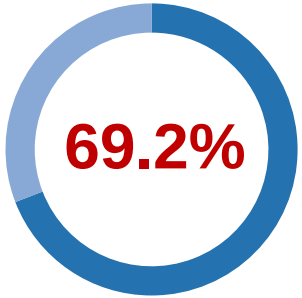


未修改的資料



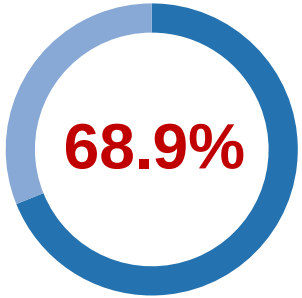
標籤化

Label Encoding



增加維度

One Hot Encoding



用敘述統計取代

Mean Encoding

報告人：陳彥蓉

建立模型

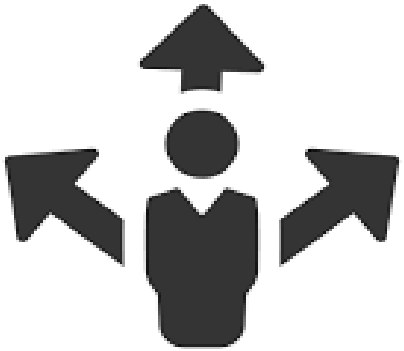
報告人：陳泓睿


MEDICAL
Health Care
Doctor
Hospital
Pharmacist
Nurse
Dentist
First Aid
Surgeon
Emergency
MEDICAL
MEDICAL
MEDICAL

建模選擇

Stacking

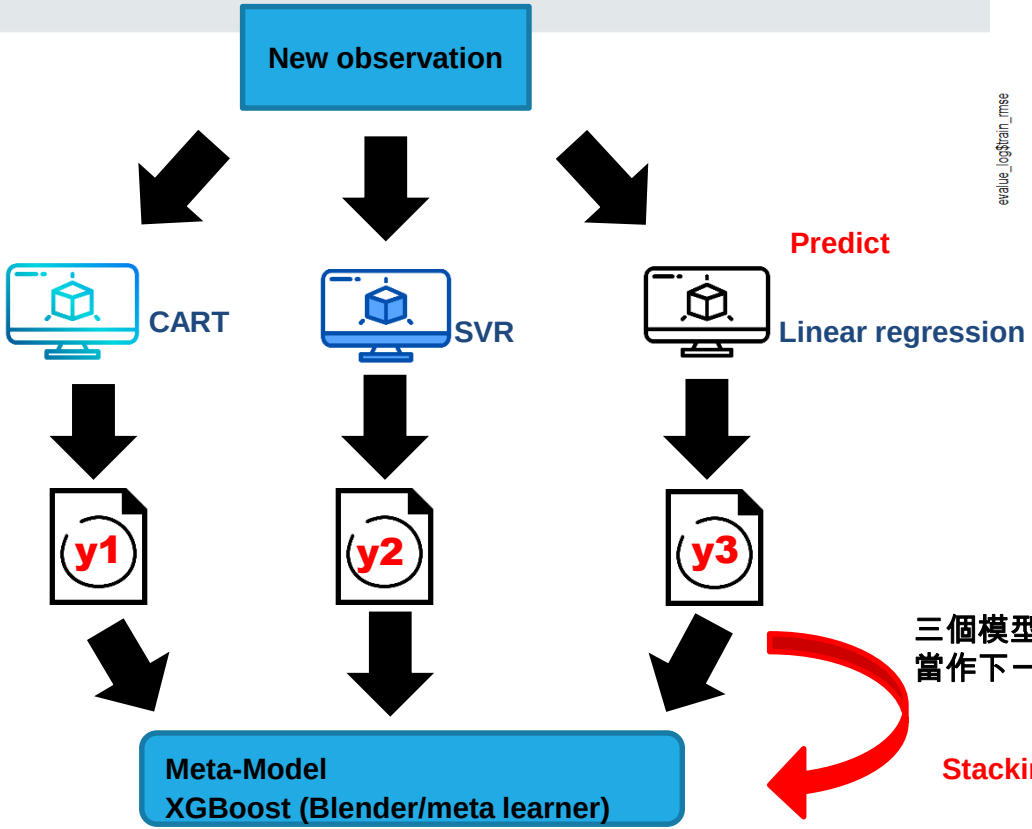
Bagging



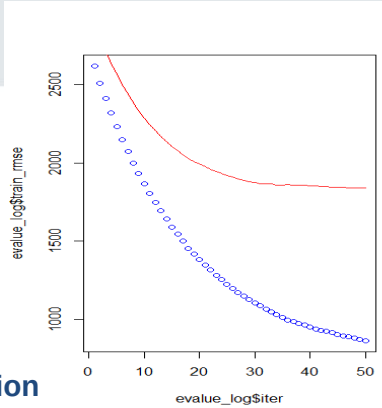
Xgboost

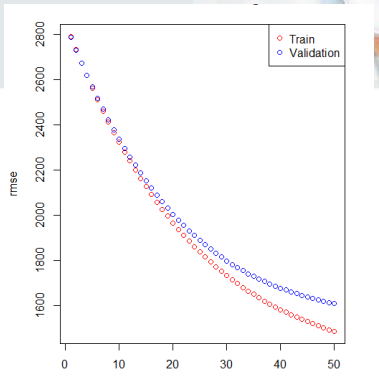
報告人：陳泓睿

Stacking



Check overfitting





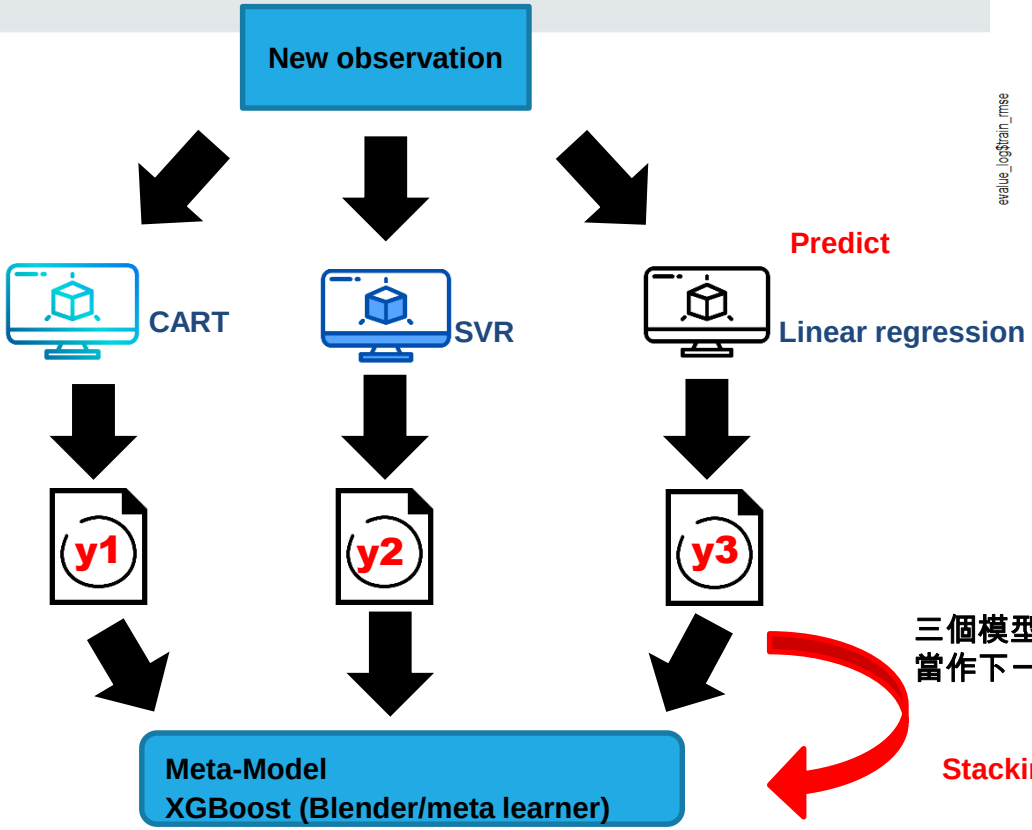

```
250 final_y = (final_1 + final_2 + final_3)/3
251 test.MAPE=mean(abs(Test$total-final_y)/Test$total)
252 cat("MAPE(test)=",test.MAPE*100,"%\n")
```

```
> xgb.model = xgb.train(paras = xgb.params,
+                       data = train_matrix,
+                       nrounds = best.nrounds)
> dtest.1 = xgb.DMatrix(data = as.matrix(meta.test.1[,1]), la
> final_1 = predict(xgb.model, dtest.1)
> dtest.2 = xgb.DMatrix(data = as.matrix(meta.test.2[,1]), la
> final_2 = predict(xgb.model, dtest.2)
> dtest.3 = xgb.DMatrix(data = as.matrix(meta.test.3[,1]), la
> final_3 = predict(xgb.model, dtest.3)
>
> final_y = (final_1 + final_2 + final_3)/3
> test.MAPE=mean(abs(Test$total-final_y)/Test$total)
> cat("MAPE(test)=",test.MAPE*100,"%\n")
MAPE(test)= 21.58811 %
```

One Hot encoding 的 MAPE

報告人：陳泓睿

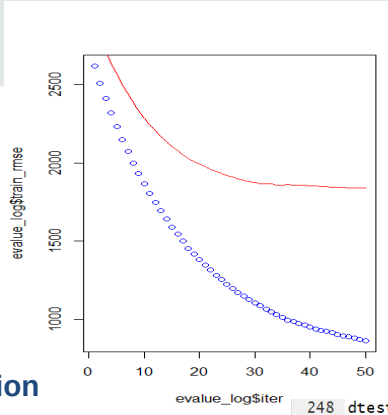
Stacking

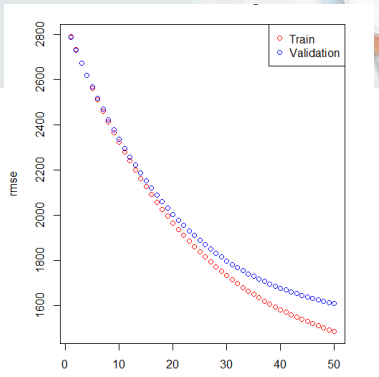


三個模型的輸出 $\hat{y}$
當作下一個模型的輸入

Stacking

Check overfitting





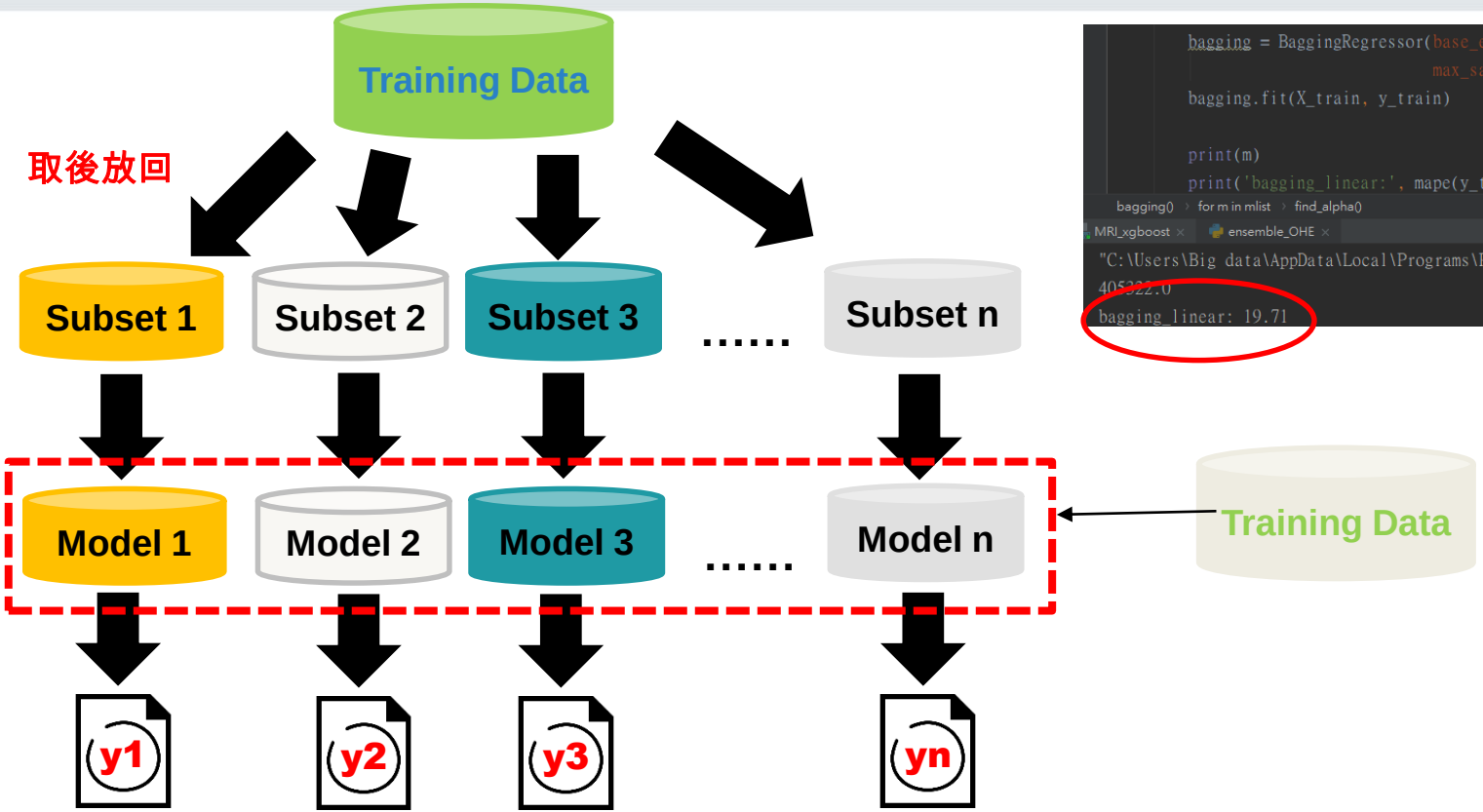

```
248 dtest.3 = xgb.DMatrix(data = as.matrix(meta.test.3[,1]),
249 final_3 = predict(xgb.model, dtest.3)
250
251 final_y = (final_1 + final_2 + final_3)/3
252 test.MAPE=mean(abs(Test$total-final_y)/Test$total)
253 cat("MAPE(test)=",test.MAPE*100,"%\n")
254
```

```
> xgb.model = xgb.train(paras = xgb.params,
+                       data = train_matrix,
+                       nrounds = best.nrounds)
> dtest.1 = xgb.DMatrix(data = as.matrix(meta.test.1[,1]), labe
> final_1 = predict(xgb.model, dtest.1)
> dtest.2 = xgb.DMatrix(data = as.matrix(meta.test.2[,1]), labe
> final_2 = predict(xgb.model, dtest.2)
> dtest.3 = xgb.DMatrix(data = as.matrix(meta.test.3[,1]), labe
> final_3 = predict(xgb.model, dtest.3)
>
> final_y = (final_1 + final_2 + final_3)/3
> test.MAPE=mean(abs(Test$total-final_y)/Test$total)
> cat("MAPE(test)=",test.MAPE*100,"%\n")
MAPE(test)= 18.57524 %
```

Mean encoding 的 MAPE

報告人：陳泓睿

Bagging



One Hot encoding 的 MAPE

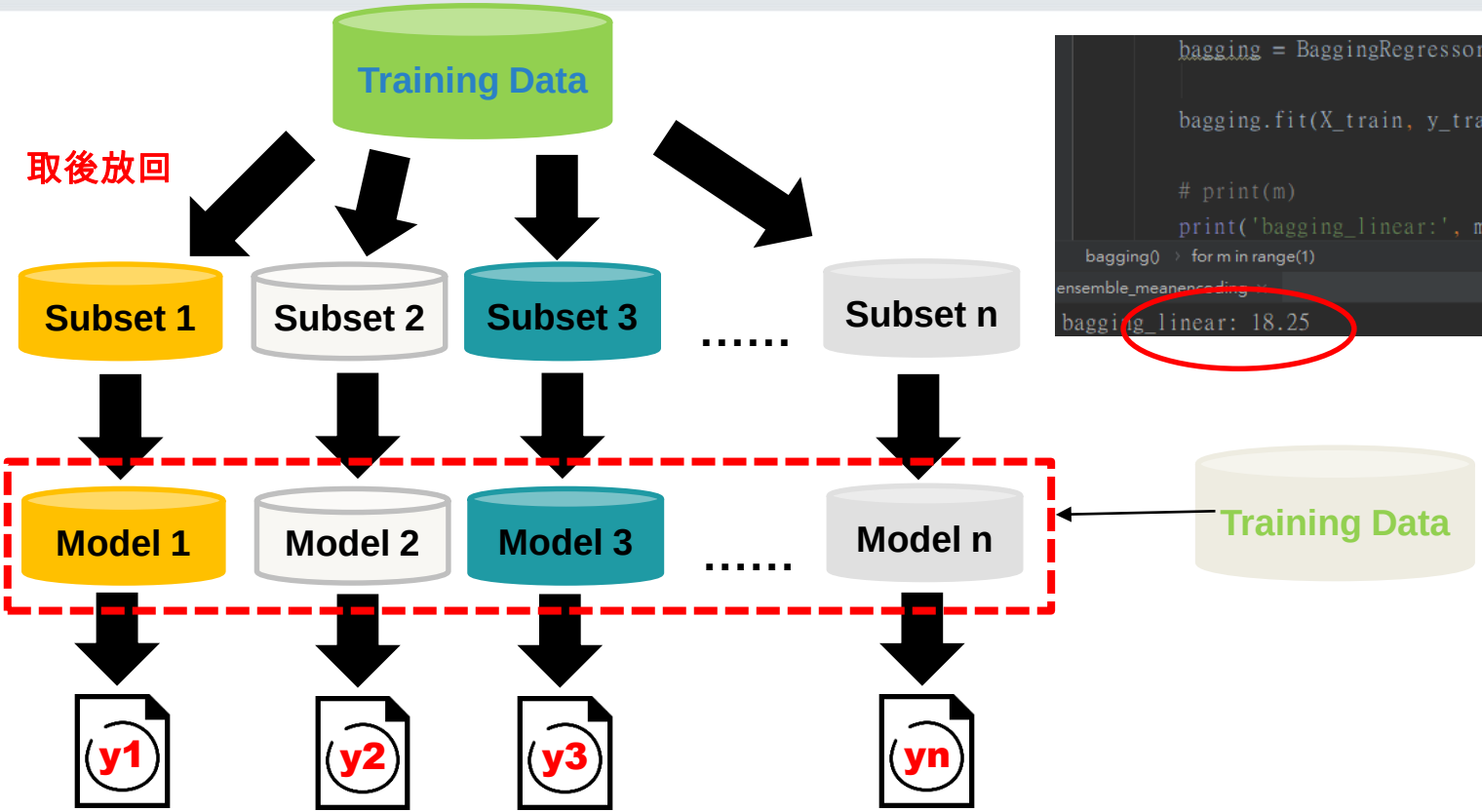
```
bagging = BaggingRegressor(base_estimator=linear_model.LinearRegression(),
                           max_samples=.1, max_features=1)
bagging.fit(X_train, y_train)

print(m)
print('bagging_linear:', mape(y_test, bagging.predict(X_test)))
```

```
bagging() › for m in mlist › find_alpha()
MRI_xgboost × ensemble_OHE ×
"C:\Users\Big data\AppData\Local\Programs\Python\Python36\python.exe" "C:/Users/Big
405322.0
bagging_linear: 19.71
```

報告人：陳泓睿

Bagging

Mean encoding 的 MAPE




```
bagging = BaggingRegressor(base_estimator=linear_model.Lin
                           max_samples=.1, max_features=1)
bagging.fit(X_train, y_train)

# print(m)
print('bagging_linear:', mape(y_test, bagging.predict(X_te
```

bagging() › for m in range(1)

ensemble_meanencoding

```
bagging_linear: 18.25
```

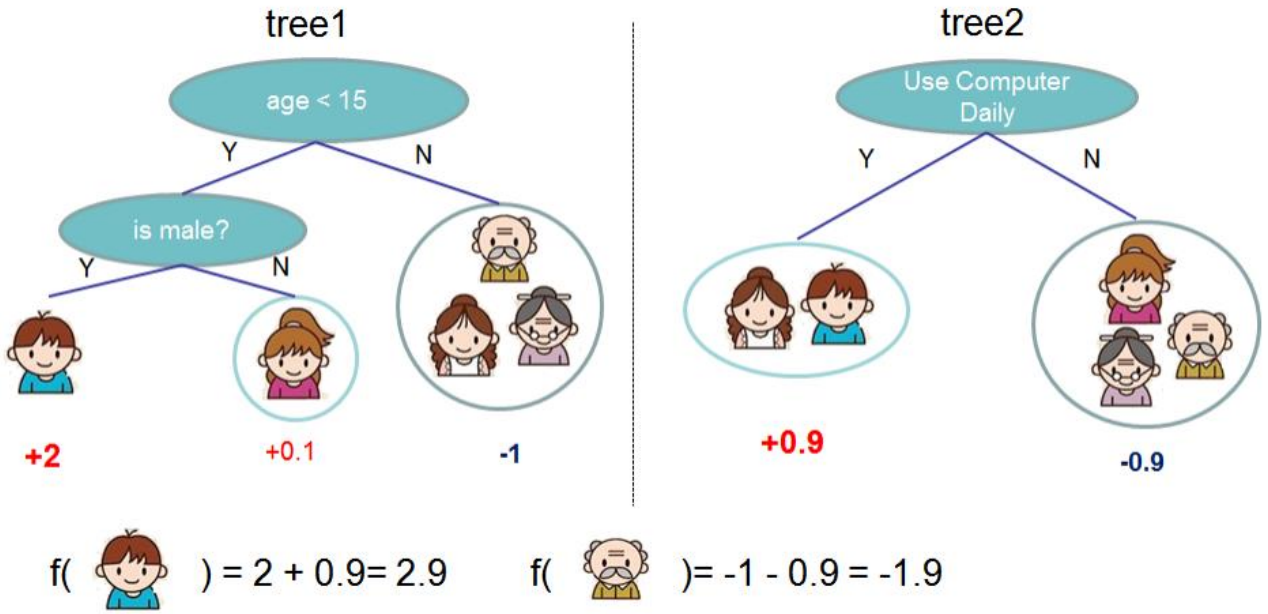
報告人：陳泓睿

Xgboost 建模

Boosting算法的思想是將
許多弱分類器集成在一起
形成一個強分類器。

適用於
連續(數值)型X欄位

解決預測與分類問題



Xgboost 調整參數

```python
mr_params_dict = {
    'n_estimators': [35,36,37,38,39,40],
    'max_depth': [3, 4, 5, 6, 7, 8],
    'min_child_weight': [ 2, 3, 4, 5, 6],
    'gamma': [0.1, 0.2, 0.3, 0.4, 0.5],
    'subsample': [0.6, 0.7, 0.8, 0.9],
    'colsample_bytree': [0.6, 0.7, 0.8, 0.9],
    'reg_alpha': [0.05, 0.1, 1, 2, 3],
    'reg_lambda': [0.05, 0.1, 1, 2, 3],
    'learning_rate': [0.01, 0.05, 0.1, 0.2]
}
other_params = {'learning_rate': 0.1, 'n_estimators': 39, 'max_depth': 4, 'min_child_weight':
                'seed': 0,'subsample': 0.6, 'colsample_bytree': 0.8, 'gamma': 0.1,
                'reg_alpha': 3, 'reg_lambda': 0.1}
gbm = xgb.XGBRegressor(**other_params)

optimized_GBM = GridSearchCV(estimator=gbm, param_grid=mr_params_dict,
                             scoring='r2', cv=5, verbose=1, n_jobs=-1)
optimized_GBM.fit(X_train, y_train)
evalute_result = optimized_GBM.cv_results_
print('引數的最佳取值：{0}'.format(optimized_GBM.best_params_))
```

```
Fitting 5 folds for each of 1440000 candidates, totalling 7200000 fits
```

進行調整XG的參數
執行後的結果顯示
引數的最佳取值

初想法:
所有結果
我全都要跑

缺點:太過耗時
調一次要等一天半

```
[Parallel(n_jobs=-1)]: Done 6400776 tasks      | elapsed: 1914.8min
[Parallel(n_jobs=-1)]: Done 6451476 tasks      | elapsed: 1930.4min
[Parallel(n_jobs=-1)]: Done 6502376 tasks      | elapsed: 1947.6min
[Parallel(n_jobs=-1)]: Done 6553476 tasks      | elapsed: 1964.0min
[Parallel(n_jobs=-1)]: Done 6604776 tasks      | elapsed: 1979.4min
[Parallel(n_jobs=-1)]: Done 6656276 tasks      | elapsed: 1998.3min
[Parallel(n_jobs=-1)]: Done 6707976 tasks      | elapsed: 2013.4min
[Parallel(n_jobs=-1)]: Done 6759876 tasks      | elapsed: 2032.0min
[Parallel(n_jobs=-1)]: Done 6811976 tasks      | elapsed: 2048.1min
[Parallel(n_jobs=-1)]: Done 6864276 tasks      | elapsed: 2065.7min
[Parallel(n_jobs=-1)]: Done 6916776 tasks      | elapsed: 2082.7min
[Parallel(n_jobs=-1)]: Done 6969476 tasks      | elapsed: 2098.5min
[Parallel(n_jobs=-1)]: Done 7022376 tasks      | elapsed: 2117.9min
[Parallel(n_jobs=-1)]: Done 7075476 tasks      | elapsed: 2133.5min
[Parallel(n_jobs=-1)]: Done 7128776 tasks      | elapsed: 2151.9min
[Parallel(n_jobs=-1)]: Done 7182276 tasks      | elapsed: 2169.6min
[Parallel(n_jobs=-1)]: Done 7200000 out of 7200000 | elapsed: 2177.1min finished
```

```
引數的最佳取值：{'colsample_bytree': 0.6, 'gamma': 0.1, 'learning_rate': 0.1, 'max_depth': 3, 'mi
n_child_weight': 6, 'n_estimators': 37, 'reg_alpha': 3, 'reg_lambda': 3, 'subsample': 0.6}
```

Xgboost 調整參數

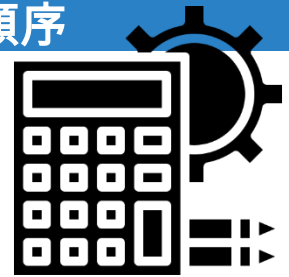
```
1  X_train, X_test, y_train, y_test = train_test_split(
2      data.drop(['total', 'PNO'], axis=1), data[['total','PNO']], test_size=0.3)
3  #
4  y_train = y_train.drop(['PNO'], axis=1)
5  #
6  X_train = X_train.as_matrix()
7  X_test = X_test.as_matrix()
8  #
9  MR_params = {'n_estimators': [35,36,37,38,39,40]}
10 MR_params = {'max_depth': [3, 4, 5, 6, 7, 8, 9, 10], 'min_child_weight': [1, 2, 3, 4, 5, 6]}
11 MR_params = {'gamma': [0.1, 0.2, 0.3, 0.4, 0.5, 0.6]}
12 MR_params = {'subsample': [0.6, 0.7, 0.8, 0.9], 'colsample_bytree': [0.6, 0.7, 0.8, 0.9]}
13 MR_params = {'reg_alpha': [0.05, 0.1, 1, 2, 3], 'reg_lambda': [0.05, 0.1, 1, 2, 3]}
14 MR_params = {'learning_rate': [0.01, 0.05, 0.1, 0.2]}
15 #調動上面修改下面 一行一行調 每調動一次請把最佳結果填入下面參數
16 other_params = {'learning_rate': 0.1, 'n_estimators': 48, 'max_depth': 5, 'min_child_weight': 1,
17                 'seed': 0,'subsample': 0.9, 'colsample_bytree': 0.6, 'gamma': 0.1,
18                 'reg_alpha': 3, 'reg_lambda': 3}
19 gbm = xgb.XGBRegressor(**other_params).fit(X_train, y_train)
20 optimized_GBM = GridSearchCV(estimator=gbm, param_grid=MR_params,
21                              scoring='r2', cv=5, verbose=1, n_jobs=4)
22 optimized_GBM.fit(X_train, y_train)
23 evalute_result = optimized_GBM.cv_results_
24 print('引數的最佳取值：{0}'.format(optimized_GBM.best_params_))
```

```
Fitting 5 folds for each of 4 candidates, totalling 20 fits
[Parallel(n_jobs=4)]: Using backend LokyBackend with 4 concurrent workers.
引數的最佳取值：{'learning_rate': 0.1}
```

依據每一次調整參數找出最佳值

進行調整XG的參數執行後的結果顯示引數的最佳取值

調整方式及順序



由9至14行
逐行調整Xgboost所需的相關參數

依序調整**最佳迭帶次數** n_estimators
最小葉子節點樣本權重和 min_child_weight
最佳深度 max_depth
除錯引數 gamma……等

在下面印出各個引述的最佳值
並記錄在other_params裡面

報告人：陳泓睿

```python
gbm = xgb.XGBRegressor(learning_rate=0.1, n_estimators=38, max_depth=4,
                       min_child_weight=5, seed=0, subsample=0.6, colsample_bytree=0.6,
                       gamma=0.1, reg_alpha=2, reg_lambda=2).fit(X_train, y_train)
predictions = gbm.predict(X_test)
```

```python
def MAPE(true, pred):

    diff = np.abs(np.array(true) - np.array(pred))
    global y_test
    y_test['pred'] = pred.astype(int)
    y_test['diff']= diff.astype(int)
    print(y_test)
    return np.mean(diff / true)
print('xgb模型的MAPE :', MAPE(y_test['total'].astype(int), predictions))
```

```
      total       PNO  pred  diff
415   3000.0  552[illegible]  3449   449
1429  2280.0  433[illegible]  2511   231
2234  2100.0   91[illegible]  2345   245
341   2100.0  517[illegible]  2473   373
4700  1500.0  361[illegible]  1990   490
2     3000.0  176[illegible]  2467   532
4165  2100.0  417[illegible]  2226   126
4015  2400.0    3[illegible]  1949   450
3719  1500.0   12[illegible]  1940   440
4761  3000.0  377[illegible]  2512   487
2623  1620.0  150[illegible]  1924   304
4194  3300.0  176[illegible]  2560   739
4633  2700.0    1[illegible]  2201   498
2477  2100.0  537[illegible]  1981   118
253   1500.0  517[illegible]  2393   893

[1624 rows x 4 columns]
xgb模型的MAPE : 0.1591764464475846
```

Xgboost建模

調校完Xgboost參數之後，
便可以代入模型，得出預測結果

Total 手寫登記時間
PNO 病歷號碼
Pred 預測時間
Diff 預測和實際的差距絕對值
單位為秒數

Mape落在20以下
效果良好

報告人：陳泓睿

Output

```
1  y_test[y_test['PNO']=='498745']#實際時間遠大於預測時間 AGE:52
```

	total	PNO	pred	diff
3229	4200.0	498745	2546	1653

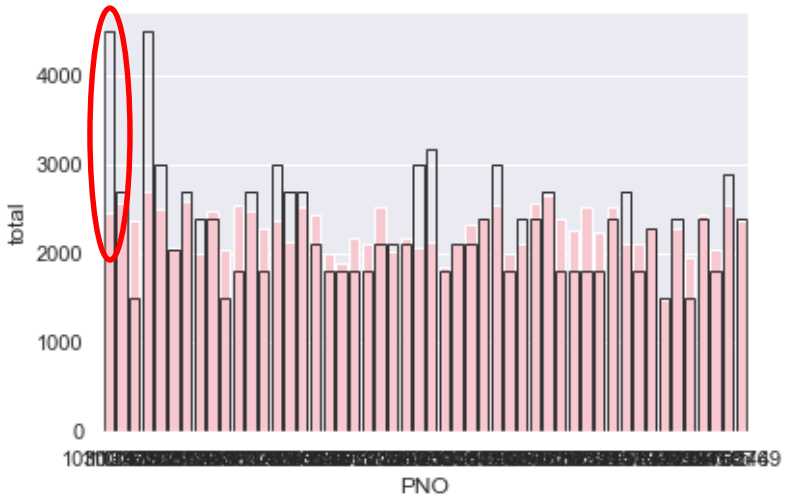
26分鐘

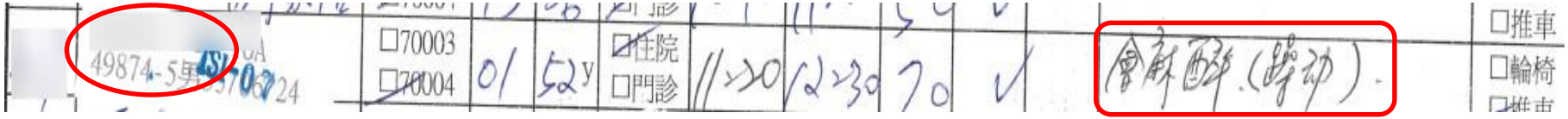
```
1  y_test[y_test['PNO']=='52327940']#同樣條件下的預測表現
```

	total	PNO	pred	diff
1843	2700.0	52327940	2520	179

3分鐘

把最終結果以圖表呈現
會發現有些病患的預測時間遠小於實際的使用時間
回去翻閱資料找原因



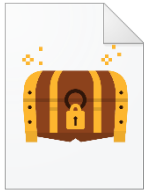


報告人：陳泓睿

Xgboost 模型輸出&測試

```
1  pickle.dump(gbm, open("pima_20190915_xgb.pickle", "wb"))
```

模型輸出後革命尚未成功
還要確保可以正常執行



pima_20190915_xgb.pickle

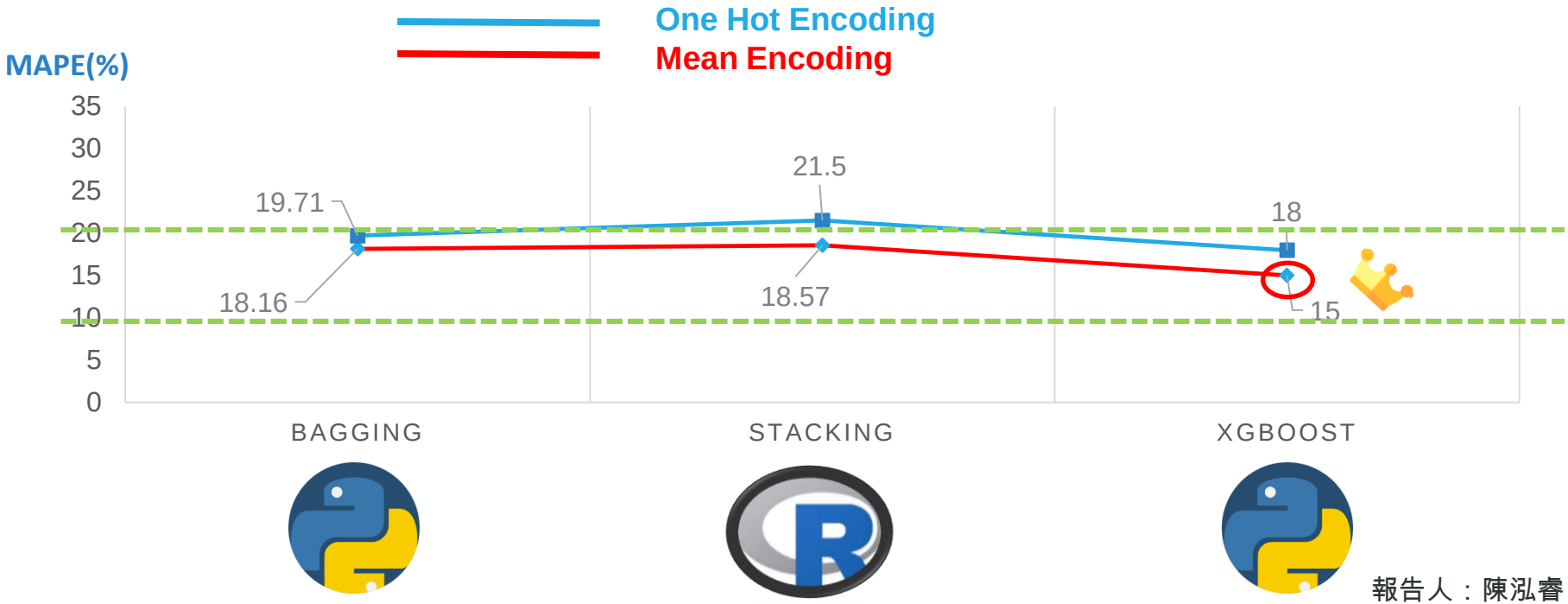
```
13      #將原變數用字典查詢其排名
14
15      select =["ITEM", "ORDERDR",
16               "DEPT", "POS", "SEX",
17               "PLACE", "IO"]
18
19      # data['ITEM'+'_n'] = data['ITEM'].apply(lambda x: ITEM[x])
20      # print(data[['ITEM_n', 'ITEM']])
21      data["DEPT"] = data["DEPT"].apply(lambda x: str(x))
22      for col in select:
23          data[col + "_n"] = data[col].apply(lambda x:eval("%s[x]" % (col)))
24
25      #選出模型需要的變數
26      select = ["ITEM_n", "ORDERDR_n","DEPT_n",
27                "SEX_n", "POS_n",
28                "PLACE_n", "IO_n"]
29      X_test = data[select]
30      X_test = X_test.as_matrix()
31      gbm = pickle.load(open("pima_20190915_xgb.pickle", "rb"))
32      #preditions為其結果
33      predictions = gbm.predict(X_test)
34      return predictions
35
36  if __name__ == '__main__':
37      # 模擬資料讀進來的時候
38      data = pd.read_csv('test.csv', encoding='CP950')
39      print(xgbtime(data))
```

```
[2185.1719 2893.9592 2351.558  2304.5544 2352.2964 2333.5083 1858.1963
 1652.0432]
```

其實調參數對於模型準確率的提高有一定的幫助，但最重要的還是要通過**資料清洗**，**欄位選擇**，**特徵融合**，**模型融合**等手段來進行改進！

報告人：陳泓睿

比較不同的演算法



報告人：陳泓睿

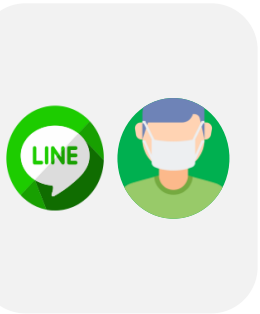
資料串流

報告人：盧冠宏


MEDICAL
Health Care
Doctor
Hospital
Pharmacist
Nurse
Dentist
First Aid
Surgeon
Emergency
MEDICAL
MEDICAL
MEDICAL

資料串流

upload



資料串流 Front-End

病患資料

Flask

web development, one drop at a time

Flask 易開發且輕量化的Python前端框架

作為客戶端的GUI使用介面 – 視覺化呈現

- 呈現敘述性統計資訊
- 輸入病患資訊
- 查詢預測時間

CSV 檔

Better Method

```
import pyspark
from pyspark import SparkConf, SparkContext

def change(record):
try:
    temp = int(record.split(",")[10])
except ValueError:
    return False
return True
```

```
sc = SparkContext()
data = sc.textFile("CSV in HDFS")
```

讀取CSV時為RDD型態, 效率更快

```
mod_data = data.filter(change)
r1 = mod_data.map(lambda x: (int(x.split(",")[i])))
r2 = r1.map(lambda x: np.array(x,dtype=int))
r3 = r2.map(lambda x: x.reshape(1,-1))
```

RDD程式邏輯 –資料型態轉
最後再做 Inference

報告人：盧冠宏

資料串流 Back-End

ORM - sqlalchemy

```python
import mysql.connector
from sqlalchemy import create_engine

def put_sqlalchemy(p):
    #conn=mysql.connector.connect(DB,IP,user,password)
    engine = create_engine('mysql+mysqlconnector://root&Password&IP&DB')
    for i in p:
        t = datetime.datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        d = {'PNO': [i[0]], 'PRETIME': [i[1]],'Time':t }
        df = pd.DataFrame(data=d)
        df.to_sql('result',con=engine,if_exists='append',index=False)
```

Python API - pyhdfs

```python
import pyhdfs

def savetohdfs(d):
    for i in d:
        client.append("RDD Output format to file in HDFS")
```

報告人：盧冠宏

資料串流 Front-End

- Pandas Data Frame處理資料

```
import pandas as pd
import numpy as np
```

Pandas DF Method

```
df = pd.DataFrame(columns=["Columns Names"])
for j in list_:
    x = j.split(',')
    dlist = ["np.int(x[j])"]
    s = pd.Series(dlist,index=["Columns Names"])
    df = df.append(s,ignore_index=True)

df["Columns Names"] = df["Columns Names"].astype(np.int64)
```

Pandas DataFrame 處理

```
df_arr = df.values
```

Array化利於RDD後續處理並適合帶入模型

```
import pyspark
from pyspark import SparkConf, SparkContext
```

Pandas DF Method

```
def batch(xs):
        yield list(xs)

sc = SparkContext()
n_partitions=11
rdd = sc.parallelize(df_arr, n_partitions).zipWithIndex()
result = rdd.mapPartitions(batch)\
            .map(lambda xs: ([x[0] for x in xs],\
                             [x[1] for x in xs]))\
            .flatMap(lambda x: zip(x[1],\
                     xgr_bc.value.predict(x[0])))
result.saveAsTextFile("HDFS output CSV/Text")
```

(RDD 1, Index = 1)
(RDD 2, Index = 2)

- Spark RDD 處理資料

```
import pyspark
from pyspark import SparkConf, SparkContext
```

Better Method

```
def change(record):
try:
    temp = int(record.split(",")[10])
except ValueError:
    return False
return True

sc = SparkContext()
data = sc.textFile("CSV in HDFS")
```

讀取CSV時為RDD型態, 效率更快

```
mod_data = data.filter(change)
r1 = mod_data.map(lambda x: (int(x.split(",")[i])))
r2 = r1.map(lambda x: np.array(x,dtype=int))
r3 = r2.map(lambda x: x.reshape(1,-1))
result = r3.map(lambda x: xgr.predict(x))
```

RDD程式邏輯
資料型態轉換
最後再做 Inference

資料串流 Issue & Solution

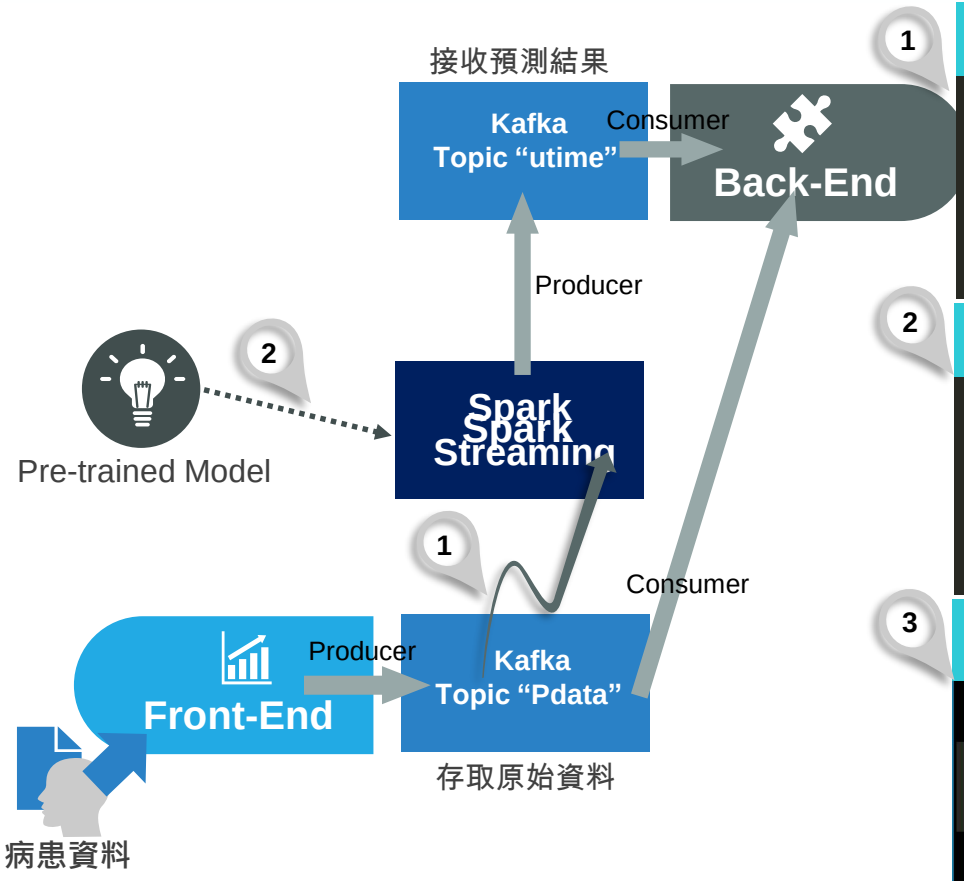
- Millions of data importing to web at same time
- Database shut down
- Overloading of database
- Data transfer Slowly

Real-time Data Pipeline

- 即時處理
- 高吞吐
- 低延遲
- 叢集

報告人：盧冠宏

資料串流 Mechanism



1

How to match with Kafka ? Consumer is unnecessary

```
from pyspark.streaming import StreamingContext
from pyspark.streaming.kafka import KafkaUtils

sc = spark.sparkContext
ssc = StreamingContext(sc, 5)
#ser consumer kafkastream take from topic  Pdata
lines = KafkaUtils.createStream(ssc, 'kafka Topic & IP')
```

2

How to import Scikit-Learn Model ?

```
from sklearn.externals import joblib

load_file = open("Model Pickle file", 'rb')
MRI_Model = joblib.load(load_file)
load_file.close()
gbr_bc = sc.broadcast(MRI_Model)
```

3

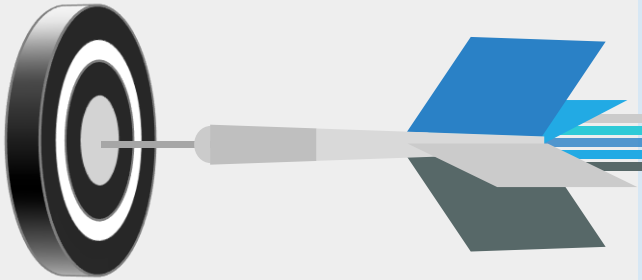
Results

```
result = Dstreams.map(lambda x: (x[0],int(xgr_bc.value.predict(x[1]))))
```

資料串流 TARGET

作動機制

- 由 Kafka 做資料傳遞
- Spark Streaming 搭配 ML Model , 監督式學習即時預測
- 即時回傳至 Web 和 Database



Real -Time Predictive Usage Time

- ***預測誤差小 (MAPE < 20 %)***
- ***資料傳遞快速 (Feedback ~ 1 Sec)***
- ***資料輕量化拿取***
- ***醫院人員利於查詢***
- ***OLAP***

Machine Learning
預測/ 輔助判斷

Spark Streaming
串流/ 運算

Kafka
傳遞/ 交換

Front-End : Flask
前端/ 客戶端(醫院人員)

Back-End : HDFS & MySQL
資料儲存/ 資料應用

報告人：盧冠宏

資料串流 實現 & 流程

02.Spark Streaming

```
time: 2019-09-20 07:47:54
time: 2019-09-20 07:47:57
('98465', 39)
('707', 30)
('883', 38)
time: 2019-09-20 07:48:00
('999', 27)
time: 2019-09-20 07:48:03
```

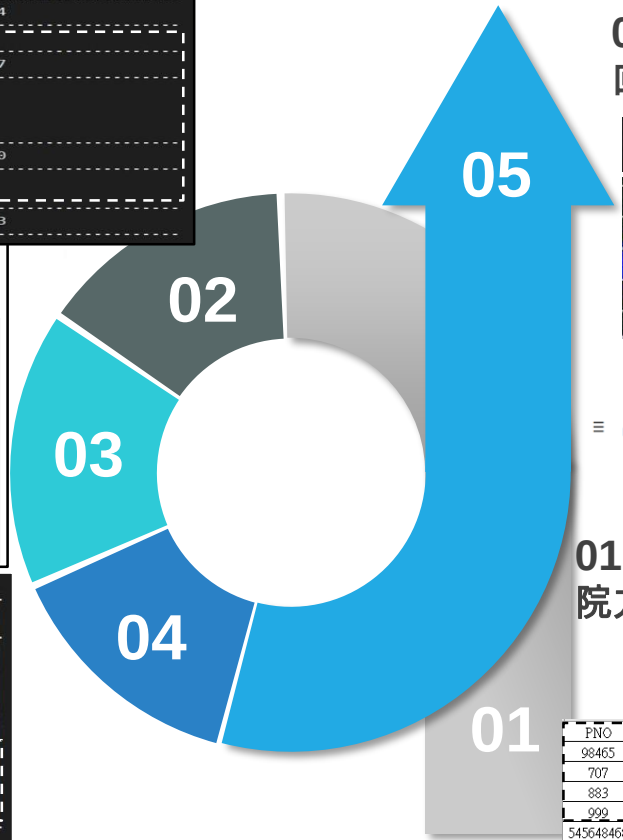
03.HDFS

Home / user / cloudera / model_deploy / output / utime.csv

```
777777,51
6666,51
None,38
3,38
23231,38
98465,39
707,30
883,38
999,27
```

04.MySQL

```
mysql> select * from result;
+------------+---------+---------------------+
| PNO        | PRETIME | Time                |
+------------+---------+---------------------+
| 3426494644 |      44 | 2019-09-16 02:08:10 |
| 1369451    |      34 | 2019-09-20 01:25:10 |
| NULL       |      38 | 2019-09-20 07:03:09 |
| 3          |      38 | 2019-09-20 07:16:17 |
| 23231      |      38 | 2019-09-20 07:47:21 |
| 98465      |      39 | 2019-09-20 07:47:59 |
| 707        |      30 | 2019-09-20 07:47:59 |
| 883        |      38 | 2019-09-20 07:48:01 |
| 999        |      27 | 2019-09-20 07:48:08 |
+------------+---------+---------------------+
9 rows in set (0.00 sec)
```



05.預測結果

回傳至Kafka 將會由Web讀取

```
bin/kafka-console-consumer.sh
The Patient NO. is 23231, need 38 minutes
The Patient NO. is 707, need 30 minutes
The Patient NO. is 98465, need 39 minutes
The Patient NO. is 883, need 38 minutes
The Patient NO. is 999, need 27 minutes
```

01.Web

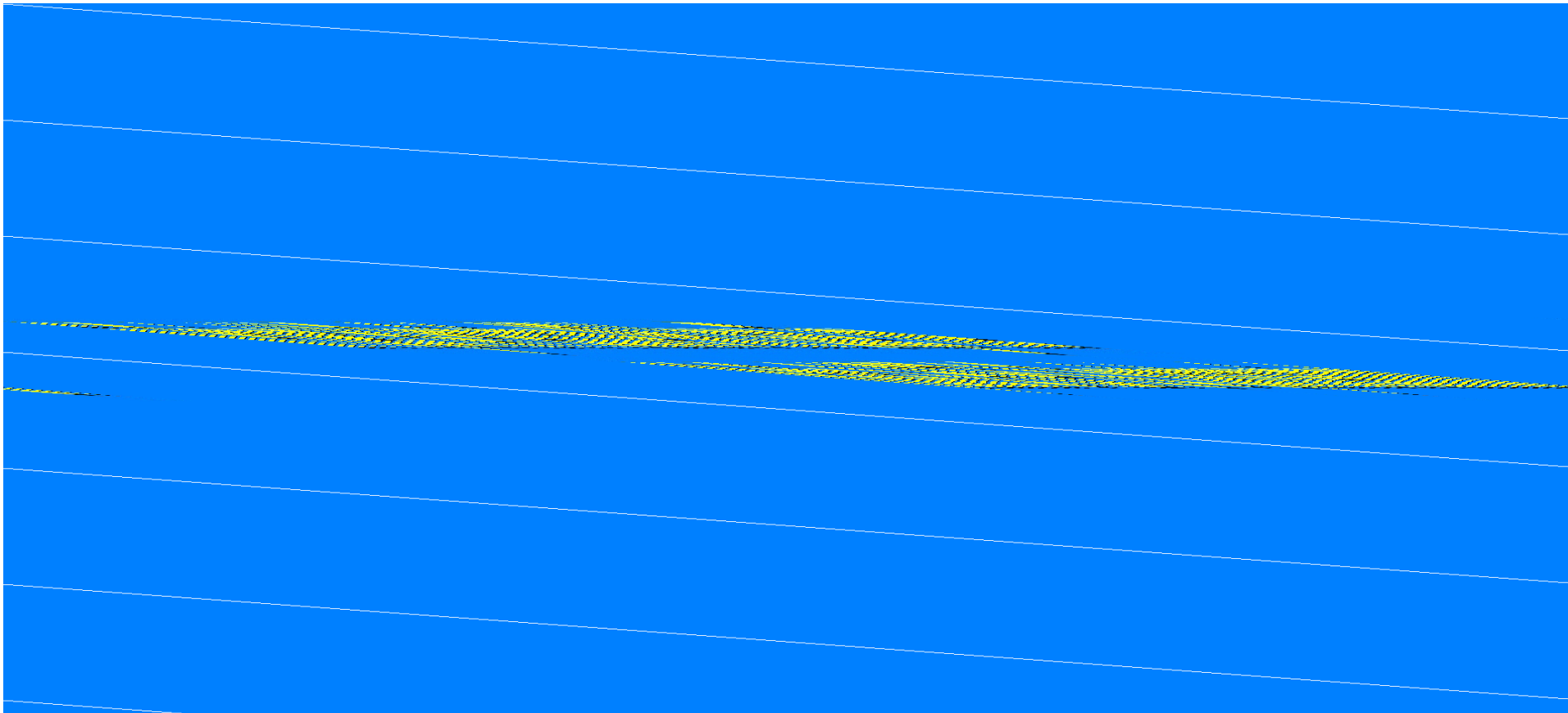
院方輸入病患資料

Hospital Assistant

病歷號碼 525252
檢查代碼 70003
醫生代碼 6379
檢查地點代碼 3T
門診急診住院代碼 O
科別代碼 13

PNO	ITEM	ORDERDR	PLACE	IO	DEPT	SEX	AGE	POS
98465	70003	4954	1.5T	O	53	F	76	32
707	70003	4368	1.5T	O	52	F	22	34
883	70003	5824	1.5T	O	20	F	68	32
999	70003	5228	1.5T	O	52	F	84	34
545648468	70003	4351	1.5T	O	52	F	43	34
515	70003	4368	1.5T	O	52	M	53	34

資料串流 Demo (Streaming & Kafka)



影像辨識

報告人：郭勇宗


MEDICAL
Health Care
Doctor
Hospital
Pharmacist
Nurse
Dentist
First Aid
Surgeon
Emergency
MEDICAL
MEDICAL
MEDICAL

醫學影像 處理流程

CT 影像

NIFTI
(3D影像)

資料
ETL

DICOM
(2D影像)

Deep
Learning

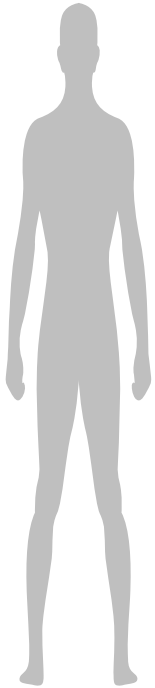
報告人：郭勇宗

醫學影像簡介

- **什麼是醫學影像？**
  - 醫學影像是反映解剖區域內部結構或內部功能的影像，經由採樣或重建產生的影像表徵的數值映射到不同空間位置上，組成2D或3D影像
- **醫學影像格式：**
  - 放射影像有6種主要的格式，分別為DICOM（醫學數位成像和通訊）、NIFTI（神經影像資訊技術）、PAR/REC（Philips磁共振掃描格式）、ANALYZE（Mayo醫學成像）、NRRD（近原始柵格資料）和MNIC。
- **DICOM和NIFTI間的區別：**
  - DICOM和NIFTI之間最主要的區別在於NIFTI中的原始影像資料是以3D影像的格式儲存的，而DICOM是以3D影像片段的格式儲存。

報告人：郭勇宗

來源資料說明



骨頭(bone)

腹部(abdomen)

縱隔 (mediastinum)

肝(liver)

肺(lung)

腎(kidney)

軟組織(soft tissue)

骨盆(pelvis)

File_ name	Patient_ index	Study_ index	Series_ ID	Key_slice_ index	Bounding_ boxes	Coarse_lesion_ type	Slice_ range	Image_ size	DICOM_ windows	Train_ Test
000001_01 _01_109.pn g	1	1	1	109	226.169, 90.0204, 241.252, 111.977	3	103, 11 5	512, 512	-175, 275	2
000001_02 _01_014.pn g	1	2	1	14	217.396, 284.296, 233.978, 310.294	4	8, 23	512, 512	-175, 275	1
000001_02 _01_017.pn g	1	2	1	17	229.412, 258.371, 285.221, 325.763	6	8, 23	512, 512	-160, 240	1

◆File_name：是由三個欄位所組成(Patient_index + Study_index + Series_ID)
◆Key_slice_index：凸顯主要病變的的切片(slice)
◆Bounding_boxes：框出病變(lesion)的位置
◆Coarse_lesion_type：病變的部位，分為8個種類(如左圖所示)
◆Slice_range：CT影像提供切片(slice)的範圍
◆Image_size：影像的尺寸
◆DICOM_windows：影像對比的調整(min~max)
◆Train_Test：分為訓練集(train)和測試集(test)

報告人：郭勇宗

File_name組成:

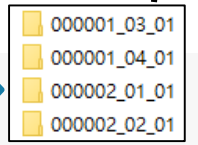
```
'Patien_index'(6位數)_'Study_index'(2位數)_'Series_ID'(2位數)_'Key_slice_index'(3位數)
"%06d" % df.iloc[0]['Patient_index'] + "_" + "%02d" % df.iloc[0]['Study_index'] + "_" + \
"%02d" % df.iloc[0]['Series_ID'] + "_" + "%03d" % df.iloc[0]['Key_slice_index'] + ".png"
```

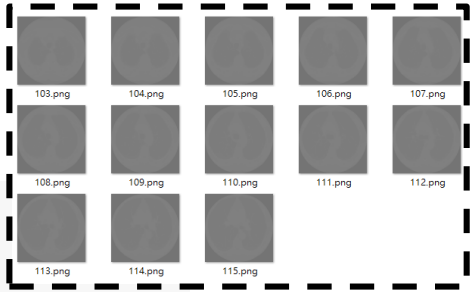
設定資料夾及NIFTI名稱

- 設定資料夾名稱格式
- 設定NIFTI(3D影像)名稱格式

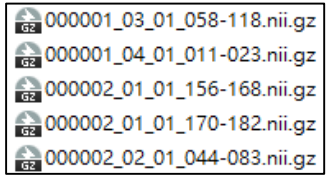
```
df = pd.read_csv('DL_info.csv')
niftiPath = 'C:/Users/Big data/Desktop/Images_nift/'
#核對 file_name 在資料夾內的順序
Images_png_Dir = 'E:/Anaconda3/Test_Image/Images_png/'
Images_png_list = os.listdir(Images_png_Dir)

for j in range(len(df)):
    folder_Name = "%06d" % df.iloc[j]['Patient_index'] + "_" + "%02d" % df.iloc[j]['Study_index'] + "_" + \
    "%02d" % df.iloc[j]['Series_ID']

    nifti_Name = "%06d" % df.iloc[j]['Patient_index'] + "_" + "%02d" % df.iloc[j]['Study_index'] + "_" + \
    "%02d" % df.iloc[j]['Series_ID'] + "_" + Slice_range_start + "-" + Slice_range_end + ".nii.gz"
    print('nifti_Name=', nifti_Name)
```

設定資料夾
名稱格式





設定NIFTI
名稱格式



報告人：郭勇宗

取檔案建立index

取出資料夾內的

所有檔案

↓

建立該資料夾內檔案的

index

↓

依據官方提供的key slice

集合其餘的slice

↓

存成2D灰階影像圖

(即DICOM圖)

↓

調整DICOM照片的對比

```
if folder_Name in Images_png_list:
    #print('folder_Name=', folder_Name)
    file_list = os.listdir(Images_png_Dir + folder_Name + '/')
    #print('file_list=', file_list)
    file_name = "%03d" % df.iloc[j]['Key_slice_index'] + ".png"
    #print('file_name=', file_name)  #str
    file_seq = file_list.index(file_name)
    #print('file_seq=', file_seq)
    #確認folder中的順序後，以File_name為主，取出slice
    slice = img_data[:, :, file_seq]
    #print('slice=', slice)
    #讀取特定slice並另存成灰階圖
    plt.imsave('E:/Anaconda3/Test_Image/CT_key_slice_image/' + df.iloc[j]['File_name'], slice, cmap='Greys_r',
               vmin=window_start, vmax=window_end)
```

報告人：郭勇宗

型態轉換

取出資料夾內的所有檔案

建立該資料夾內檔案的index

依據官方提供的key slice集合其餘的slice

存成2D灰階影像圖(即DICOM圖)

調整DICOM照片的對比

```
#讀取特定slice並另存成灰階圖
plt.imsave('E:/Anaconda3/Test_Image/CT_key_slice_image/' + df.iloc[j]['File_name'], slice, cmap='Greys_r',
           vmin=window_start, vmax=window_end)
```

形態轉換

限定整數型態

```
window_start = pd.to_numeric(df.iloc[j]['DICOM_windows'].split(', ')[0]).astype(np.int64)
window_end = pd.to_numeric(df.iloc[j]['DICOM_windows'].split(', ')[1]).astype(np.int64)
```

數值型態

1. 字串型態
2.
3. 整數

報告人：郭勇宗

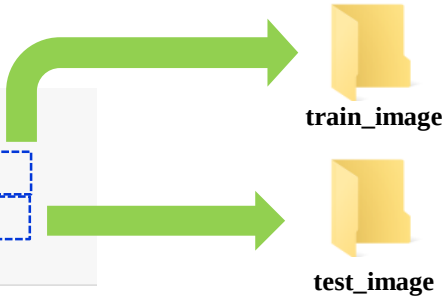
對比調整	有	無
DICOM影像		

深度學習前處理

◆ 深度學習所需的檔案：訓練集圖像、病變注解檔

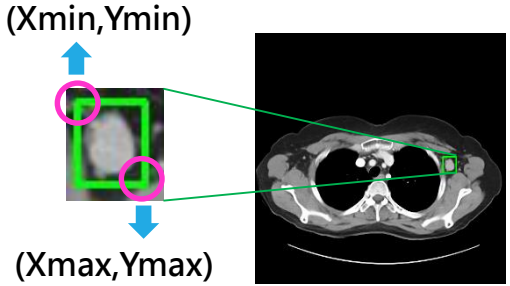
訓練集影像分類:

```
for i in range(len(df)):
    key_slice_img = cv2.imread(key_slice_image_Dir + df.iloc[i]['File_name'])
    Train_Val_Test_type = df.iloc[i]['Train_Val_Test']
    if Train_Val_Test_type == 1:
        cv2.imwrite( key_slice_image_class_Dir + 'train_image/' + df.iloc[i]['File_name'], key_slice_img)
    elif Train_Val_Test_type == 2:
        cv2.imwrite( key_slice_image_class_Dir + 'test_image/' + df.iloc[i]['File_name'], key_slice_img)
    else:
        cv2.imwrite( key_slice_image_class_Dir + 'other/' + df.iloc[i]['File_name'], key_slice_img)
```



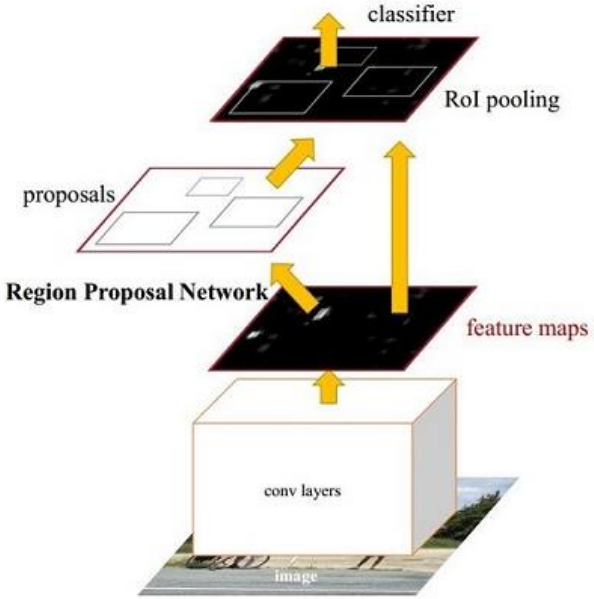
病變注解檔生成: *(格式：路徑/File_name,Xmin,Ymin,Xmax,Xmin,type)*

```
count = len(open('DL_info.csv','r').readlines())
for i in range(count):
    Xmin = pd.to_numeric(train_df.iloc[i]['Bounding_boxes'].split(', ')[0]).astype(np.int64)
    Ymin = pd.to_numeric(train_df.iloc[i]['Bounding_boxes'].split(', ')[1]).astype(np.int64)
    Xmax = pd.to_numeric(train_df.iloc[i]['Bounding_boxes'].split(', ')[2]).astype(np.int64)
    Ymax = pd.to_numeric(train_df.iloc[i]['Bounding_boxes'].split(', ')[3]).astype(np.int64)
    #print('Xmin={}, Ymin={}, Xmax={}, Ymax={}'.format(Xmin, Ymin, Xmax, Ymax))
    train_data = ('train_image/' + train_df.iloc[i]['File_name']+','+str(Xmin)+','+str(Ymin)+ ','+str(Xmax)+','+str(Ymax)+','
                  +str(train_df.iloc[i]['Coarse_lesion_type']))
```



報告人：郭勇宗

深度學習 Faster-R-CNN 簡介



◆ 檢測速度對比

	R-CNN	Fast R-CNN	Faster R-CNN
Test time per image	50 seconds	2 seconds	0.2 seconds
Speedup	1x	25x	250x

◆ Faster-R-CNN算法由兩大模塊組成：

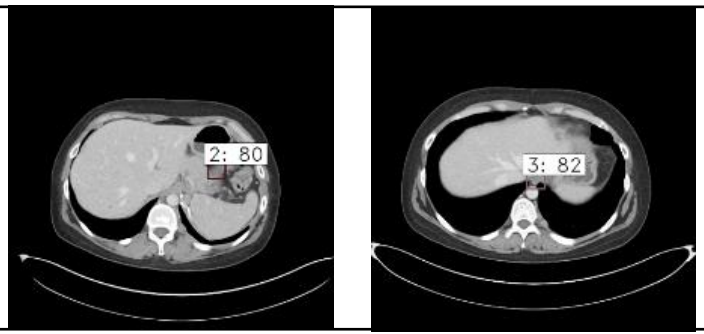
1. PRN候選框提取模塊；
2. Fast R-CNN檢測模塊。

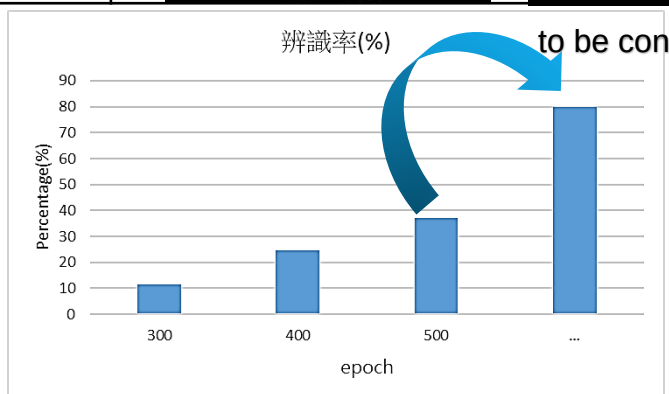
其中，RPN是全卷積神經網絡，用於提取候選框；Fast R-CNN基於RPN提取的proposal檢測並識別proposal中的目標。

訓練集影像 + 病變注解檔 + Faster-R-CNN = 模型(Model)

報告人：郭勇宗

深度學習 測試後結果

DICOM 影像調整依據	epoch [註1]	辨識率(%)	Elapsed time of model training (hr)	影像
--	300	0	67.0	--
Official	300	11.38 % (423 pcs/ 3716 pcs)	67.2	
Official	400	24.70 % (918 pcs/ 3716 pcs)	78.6	
Official	500	37.22 % (1383 pcs/ 3716 pcs)	91.8	
Official	● ● ●			



註1:
epoch:每一個完全樣本要看幾遍

報告人：郭勇宗

LineBot -推薦系統

報告人：鄧鈺蓉 & 李建德

Line bot

確認預約

- 輸入病患編碼後，查詢預約

醫院信息

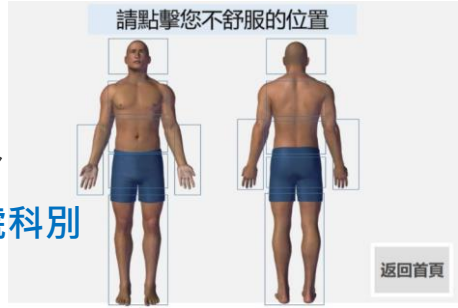
- 一鍵撥打醫院電話
- 一鍵導航至醫院地址

文章推薦

- 按照使用者情況推薦文章

協助就診

- 點擊不適的部位，建議掛號科別



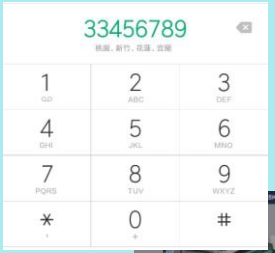
報告人：鄧鈺蓉

Line bot 醫院信息 &協助就診

醫院信息

Line URI 功能：

- 一鍵撥打醫院電話
- 一鍵導航至醫院地址



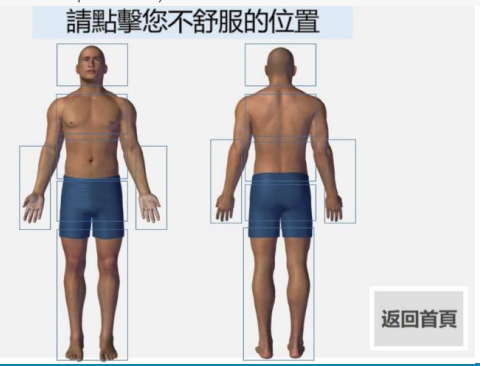

```
reply_message_info = [
    TemplateSendMessage(
    alt_text='Buttons template',
    template=ButtonsTemplate(
        thumbnail_image_url="https://%s/image/hospital.jpg"% server_url,
        title='資業會醫院',
        text='台北市大安區和平東路二段106號',
        actions=[
          {
            "type": "uri",
            "label": "**撥打** 醫院電話",
            "uri": "tel://33456789"
          },
          {
            "type": "uri",
            "label": "**導航** 醫院地址",
            "uri": "line://app/1599834661-1RXo0B54"
```

協助就診

Line RichMenu 轉場功能：

報告人：鄧鈺蓉

```
def add_richmanu(user_id,rich_menu_id):
    linkRichMenuId=secretFileContentJson.get(rich_menu_id)
    linkMenuEndpoint='https://api.line.me/v2/bot/user/%s/richmenu/%s' % (user_id, linkRichMenuId)
    linkMenuRequestHeader={'Content-Type':'image/jpeg','Authorization':'Bearer %s' % secretFileContentJson["channel_access_token
    lineLinkMenuResponse=requests.post(linkMenuEndpoint,headers=linkMenuRequestHeader)
```



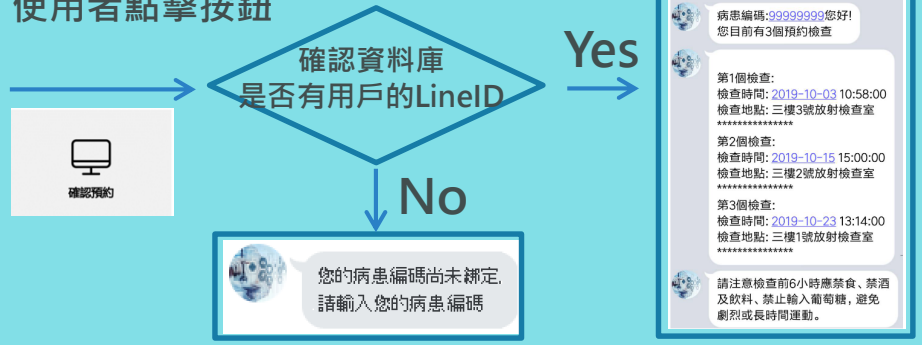

```
@handler.add(PostbackEvent)
def handle_post_message(event):
    if "::test::" in event.postback.data:
        line_bot_api.reply_message(
        event.reply_token,
        template_message_dict.get(event.postback.data))
```


```
template_message_dict = {
    "[::test::]醫院信息":reply_message_info,
    "[::test::]頭部":TextSendMessage(text="建議您掛：一般內科、家庭醫學科、神經內科、眼科、耳鼻喉科、皮膚科、精神科。"),
    "[::test::]胸部":TextSendMessage(text="建議您掛：心臟內科、胸腔內科、一般內科、家庭醫學科。"),
    "[::test::]手部":TextSendMessage(text="建議您掛：骨科、復健科、神經內科、神經外科、家庭醫學科、皮膚科。"),
    "[::test::]腿部":TextSendMessage(text="建議您掛：骨科、復健科、神經內科、神經外科、家庭醫學科、皮膚科、過敏免疫風濕科。"),
    "[::test::]腹部":TextSendMessage(text="建議您掛：胃腸科、一般內科、家庭醫學科、一般外科、腎臟科、疼痛控制科。"),
    "[::test::]生殖部":TextSendMessage(text="建議您掛：婦產科、泌尿科、腎臟科。"),
    "[::test::]背部":TextSendMessage(text="建議您掛：骨科、復健科、心臟內科、胸腔內科、神經內科、神經外科、疼痛控制科。"),
    "[::test::]臀部":TextSendMessage(text="建議您掛：一般內科、胃腸科、直腸外科、家庭醫學科。"),
    "[::test::]腰部":TextSendMessage(text="建議您掛：復健科、骨科、神經內科、疼痛控制科、腎臟科。")
```

Line bot 確認預約

報告人：鄧鈺蓉

- 使用者點擊按鈕

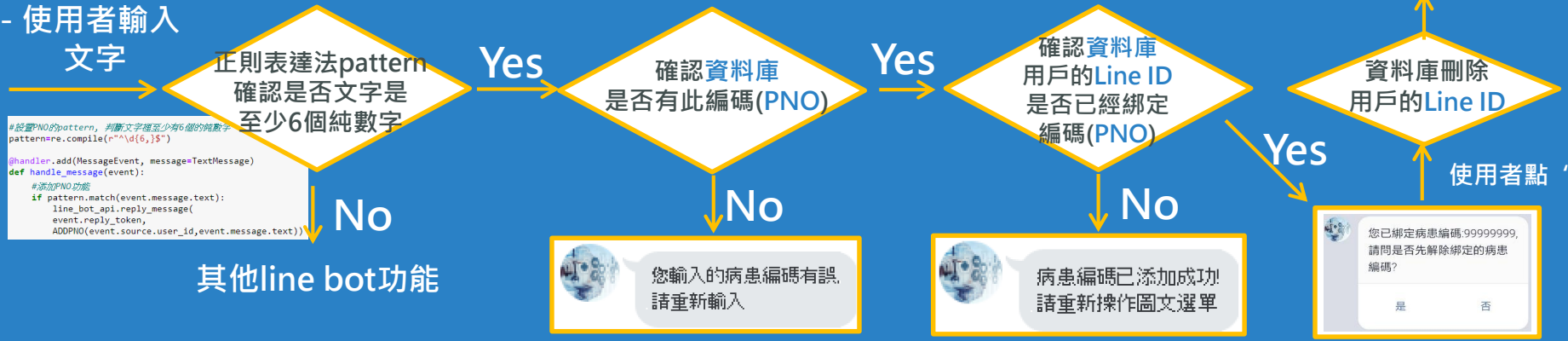


資料庫（Web醫生端建立預約）：

Line ID　　PNO

lineID	PNO	AGE	SEX	PLACE
1	11111111	66	女	三樓1號放射檢查室
1	11111111	66	女	三樓2號放射檢查室
1	22222222	82	女	三樓3號放射檢查室
1	22222222	82	女	三樓2號放射檢查室
1	22222222	82	女	三樓1號放射檢查室
Ubc2d78f9e714ca16172ef2e74326aa16	99999999	63	女	三樓2號放射檢查室
Ubc2d78f9e714ca16172ef2e74326aa16	99999999	63	女	三樓3號放射檢查室
Ubc2d78f9e714ca16172ef2e74326aa16	99999999	63	女	三樓1號放射檢查室

- 使用者輸入文字




```
#設置PNO的pattern, 判斷文字裡至少有6個的純數字
pattern=re.compile(r"^\d{6,}$")

@handler.add(MessageEvent, message=TextMessage)
def handle_message(event):
    #添加PNO功能
    if pattern.match(event.message.text):
        line_bot_api.reply_message(
        event.reply_token,
        ADDPNO(event.source.user_id,event.message.text))
```

Line bot 關鍵字查詢

將用戶輸入的文字
進行jieba分詞

尋找符合的
文章title和文章tag

成功！
文章很貼合
使用者的問題！

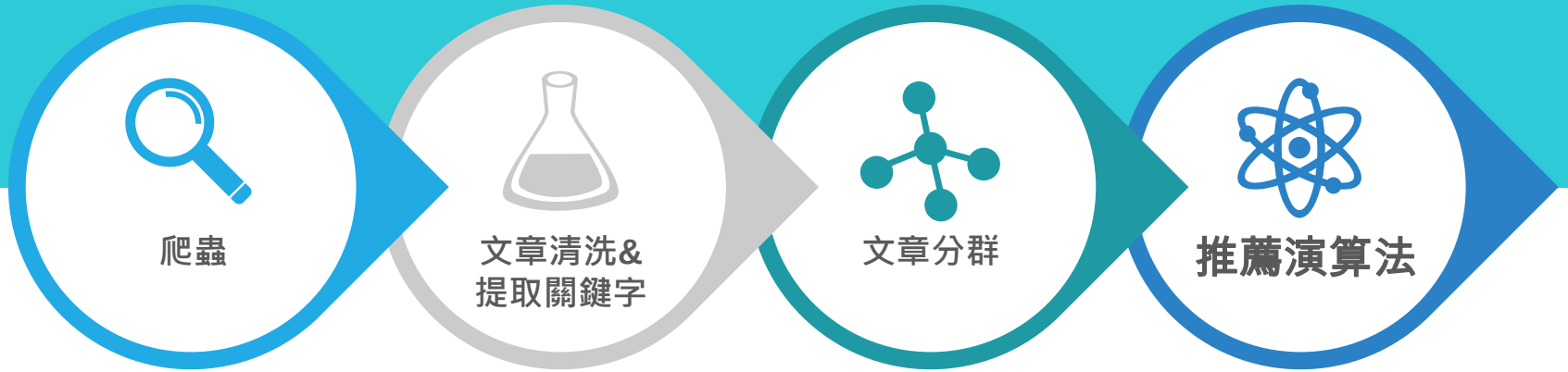
報告人：鄧鈺蓉

將此關鍵字查詢功能
作為彩蛋保留

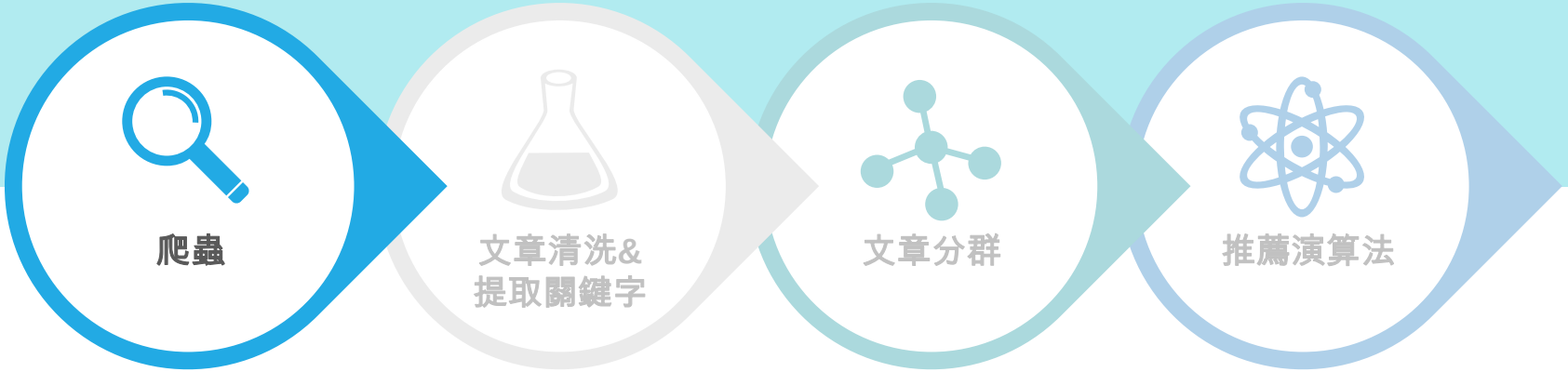
精益求精：

一. 這屬於關鍵字查詢，不是推薦的演算法。
二. 要使用者主動輸入文字，不符合用戶使用習慣。
三. 在已經掌握病患資料的情況下，能否做出量身定做的推薦

文章推薦流程



爬蟲



報告人：李建德

分析網頁

早安健康

YAHOO! 奇摩 健康

康健 For a better life

報告人：李建德

- 網址

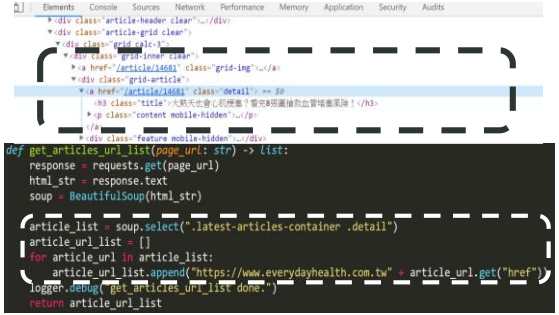
everydayhealth.com.tw/category/51/index/1

報告人：李建德

- 文章類別

健康 | 飲食 | 瘦身 | 運動 | 美麗 | 專欄 | 影音 | 護口腔 | 足健康

- 文章網址



```
def get_articles_url_list(page_url: str) -> list:
    response = requests.get(page_url)
    html_str = response.text
    soup = BeautifulSoup(html_str)

    article_list = soup.select(".latest-articles-container .detail")
    article_url_list = []
    for article_url in article_list:
        article_url_list.append("https://www.everydayhealth.com.tw" + article_url.get("href"))
    logger.debug("get_articles_url_list done.")
    return article_url_list
```

```
def get_pages_url_list(category_idx: int) -> list:
    base_url = "https://www.everydayhealth.com.tw/category/" + str(category_idx) + "/index/"
    response = requests.get(base_url+"1")
    html_str = response.text
    soup = BeautifulSoup(html_str)

    final_page_url = soup.select(".disabled+ .disabled a")[0].get("href")
    pattern = re.compile(r'[0-9]+$')
    page = pattern.findall(final_page_url)[0]
    page_url_list = [base_url+str(i) for i in range(1, int(page)+1)]
    logger.debug("get_pages_url_list done.")
    return page_url_list
```

- 頁碼

1 2 3 4 5 6 > >>

分析內文

everydayhealth.com.tw/article/22533

高血壓傷腦，失智風險達1.7倍！3類型高血壓該如何改善？

2019-09-12　新聞中心張承宇

2.9 K　收藏1

【早安健康／張承宇報導】高血壓帶來的問題。除了我們熟知的動脈硬化、中風、心肌梗塞以外，近年研究更發現，高血壓還可能會傷害腦部掌管記憶功能的海馬迴，提高失智風險，一定要積極預防改善。日本高血壓專業醫師分析了3種不同成因的高血壓，並提供了不同的預防辦法，趕快從現在開始積極改善！

第一頁　1　2　>　下一頁

需要欄位:

- 網址
- 標題
- 日期
- 作者
- 標籤
- 觀看次數
- 文章內容

```python
def parse_article(article_url: str) -> list:
    response = requests.get(article_url)
    html_str = response.text
    soup = BeautifulSoup(html_str)
    need_crawl = {  "title": ".article .title",
                    "publication_date": ".date",
                    "author": ".autor",
                    "tags": ".tag-list",
                    "read_num": "span.number span",
                    "content": "#article_page"  }
    key_list = ["title", "publication_date", "author", "tags", "read_num", "content"]
    result = []
    for key in key_list:
        result.append(soup.select(need_crawl[key])[0].text)
    # 如果文章有多頁，再次呼叫此函式下載分頁文章，並將內容串接
    if soup.select("li+ li .actbtn") != []:
        next_page_url = soup.select("li+ li .actbtn")[0].get("href")
        if next_page_url is not None and next_page_url != "javascript:void(0)":
            next_page_result = parse_article(next_page_url)
            result[5] += next_page_result[5]
    # 正規表示法匹配URL最後的數字，單頁文章URL最後數字通常有五位數
    # 有多頁的文章最後數字為分頁數，通常都是個位數，設定99來判斷是否應該存檔
    if int(re.search("[0-9]*$", article_url).group(0)) > 99:
        result.append(article_url)
    return result
```

mongoDB

報告人：李建德


```
"article_url" : "https://www.everydayhealth.com.tw/article/22533"
"_id" : ObjectId("5d83793211b6137c93118ee8"),
"title" : "高血壓傷腦，失智風險達1.7倍！3類型高血壓該如何改善？",
"publication_date" : "2019-09-12",
"author" : "新聞中心張承宇",
"tags" : "\r\n關鍵字：\r\n\r\n養生保健\r\n高血壓\r\n預防失智\r\n壓力\r\n\r\n",
"read_num" : "2.9 K",
"content" : "【早安健康／張承宇報導】高血壓帶來的問題。除了我們熟知的動脈硬化、中
更發現，高血壓還可能會傷害腦部掌管記憶功能的海馬迴，提高失智風險，一定要積極預防改
```

分析網頁

- 有文章關鍵字

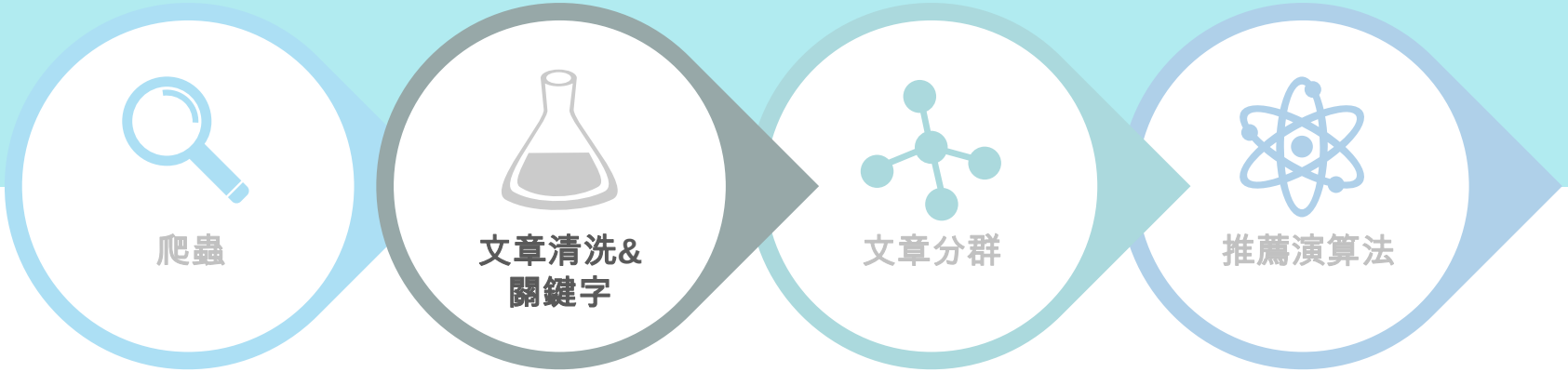
- 沒有文章關鍵字

- 網頁的文章分類

- 有些文章沒有標籤
- 不同網站分類不同
- 分類不是基於疾病

需要另外做文章分群

文章清洗&關鍵字



報告人：鄧鈺蓉

文章清洗 關鍵字

報告人：鄧鈺蓉

目標

- 將現有的標籤做成字典，給沒有標籤的文章貼標籤
- 剔除跟醫療不相關的文章跟邊角文章(如財經、除臭等內容)

- 根據網站文章分群
將不符合的文章剔除

養生保健
健康飲食
健康小撇步
保健新聞
生活智慧
生活便利貼
對抗老化
其他身心健康
清潔/打掃
穴道導引

- 刪除邊角文章：
至少要擁有一個在所有
文章中出現20次的關鍵字

```
Series=pd.Series(tfidf_vectorizer.vocabulary_)
Series[Series.values>20]. sort_values()
```

三高	21
上半身	22
上火	23
上班族	24
下半身	25
下垂	26
下巴	27
...	
鼻塞	1255
鼻子	1256
鼻水	1257
鼻炎	1258
鼻竇炎	1259
鼻腔	1260

早安健康

- 文章數：3869
- 標籤數：1831

標籤製作成字典

- 刪除邊角文章：
文章內容至少要含有3個標籤

早安健康

YAHOO! 奇摩 健康

文章數：7440
標籤數：1831

康健 For a better life

title	KW
《黃帝內經》冬季養護陽氣睡眠術，從此不失眠	失眠 睡眠 冬天 泡腳 按摩
「0負擔」降血壓！簡單起立坐下就能做到的降壓密技	降血壓 促進血液循環 動脈硬化
「不疲勞用眼法」！這樣用3C眼睛不過勞	改善視力 視力 電腦 手機 疲勞 眼睛疲勞 葉黃素
「五體散步術」讓90歲作家健檢指標All Pass	抗老 老化 老年痴呆 老花眼 衰老 散步 銀髮族 高齡 減重
「冷飲症候群」加快2倍失智速度！喝溫茶防凸肚肥胖	失智 小腹 肥胖 茶 阿茲海默症

文章清洗 關鍵字

- 關鍵字轉成次數矩陣

```
from sklearn.feature_extraction.text import CountVectorizer
vec = CountVectorizer()
tfidf_matrix = vec.fit_transform(h_content["KW"])
```

	title	URL	keywords
0	三高世代，慢性病是必然？蔬果多酚降風險，看懂哪些食物適合你	https://www.everydayhealth.com.tw//article/22269	健康飲食 腦中風 心臟病 咖啡 蔬果 花青素 大豆異黃酮
1	血型與罹癌風險有關？研究：2種血型容易得胃癌、胰臟癌	https://www.everydayhealth.com.tw//article/22350	保健新聞 血型 基因 抗體 健康飲食 癌症 中風
2	睡飽還是累？還清睡眠債的5練習，別讓昨日疲勞積到今天	https://www.everydayhealth.com.tw//article/22361	養生保健 睡眠不足 阿茲海默症 糖尿病 心血管疾病 睡眠策略
3	寒氣是子宮的毒素！必吃3蔬菜防宮寒、荷爾蒙失調	https://www.everydayhealth.com.tw//article/14927	健康飲食 子宮 生薑 薑黃 大蒜 虛寒 手腳冰冷
4	研究：腦過勞釀失智，錢包零錢爆滿竟是失智前兆！	https://www.everydayhealth.com.tw//article/20397	養生保健 失智 失智症 預防失智 疲勞 滑手機 Omega3
5	漏尿，是因為尿急、還是大笑？中醫「對症3招」緩解漏尿困擾	https://www.everydayhealth.com.tw//article/22374	保健新聞 尿失禁 漏尿 頻尿 中醫

報告人：鄧鈺蓉

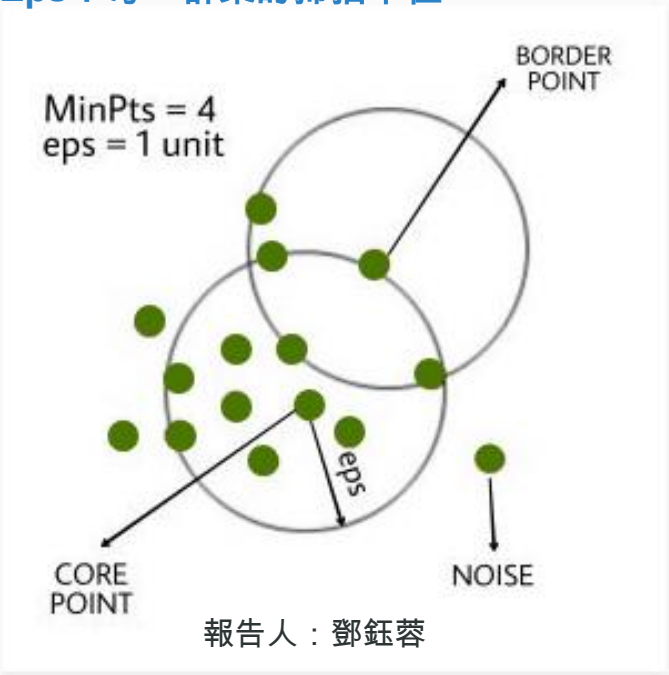
	cpr	dha	epa	gi值	omega3	o型腿	pm2	一氧化氮	三酸甘油酯	三陰交	...	黑豆水	鼻塞	鼻子	鼻子過敏	鼻炎	鼻竇炎	鼻腔	鼻血	龍眼
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7438	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
7439	0	0	0	0	0	0	0	0	0	0	...	0	1	1	0	0	0	0	0	0
7440	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0

文章分群

報告人：鄧鈺蓉

文章分群 DBSCAN

- 設定參數：
  min_samples：定義每一群集最小樣本數
  Eps：每一群集的掃描半徑



報告人：鄧鈺蓉

- 設定一：

```
1 from sklearn.cluster import DBSCAN
2 y_pred = DBSCAN(eps=1, min_samples=4).fit_predict(tfidf_matrix)
3 y_pred=pd.Series(y_pred).value_counts()
4 y_pred
```

```
-1     5640
 2      501
 1       27
 3       19
 0       17
20       16
 6       15
 4       14
11       14
18       14
 9       12
 7       12
21       12
 5       11
24       10
22        9
17        8
 8        8
12        8
        ...
15        4
28        4
16        3
dtype: int64
```

← Noisy samples are given the label -1

- 設定二：

```
1 from sklearn.cluster import DBSCAN
2 y_pred = DBSCAN(eps=1.2, min_samples=4).fit_predict(tfidf_matrix)
3 y_pred=pd.Series(y_pred).value_counts()
4 y_pred
```

```
 0    7407
-1      33
dtype: int64
```

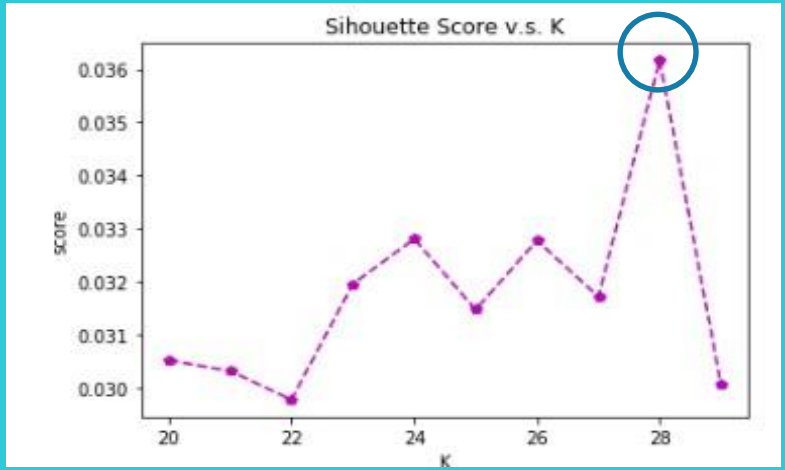
問題總結：

1. 資料集的變數過多，分群效果差
2. 被歸類為Noise的樣本數太多

文章分群　Kmeans 確定分群數量

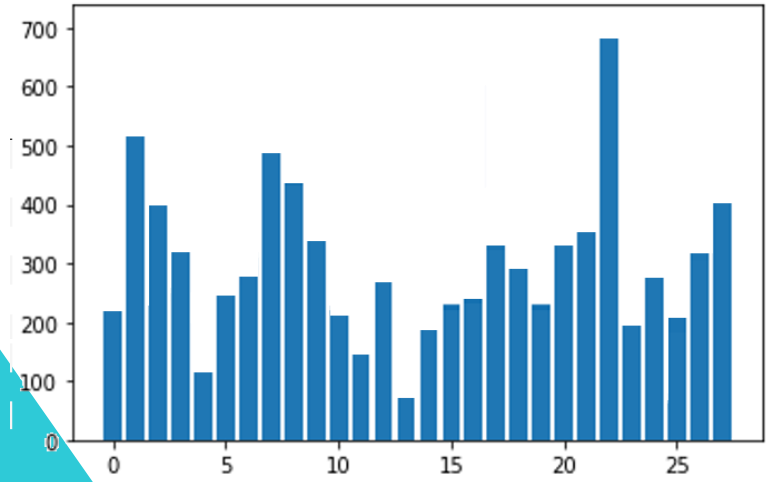
silhouette_score()
確認文章要分成幾群

```
ks = []
scores = []
for k in range(20,30):
    clu = KMeans(n_clusters=k, n_jobs=-1)
    clu.fit(tfidf_matrix)
    ks.append(k)
    scores.append(silhouette_score(tfidf_matrix, clu.labels_))
```



分成28群

- 每群的文章數量相對平均



報告人：鄧鈺蓉

文章分群　Kmeans 檢驗分群效果

- 每群出現最多的關鍵字

group	key	group	key
0	神經	14	血液循環
1	減重	15	手腳冰冷
2	高血壓	16	骨質疏鬆
3	睡眠	17	喝水
4	暈眩	18	抗老
5	穴道按摩	19	瘦身
6	運動	20	膽固醇
7	肝臟	21	失智症
8	感冒	22	骨盆
9	經痛	23	壓力
10	頭痛	24	疲勞
11	便祕	25	血糖
12	腰痛	26	眼睛
13	癌症	27	減重

- 同樣是減重的分群，文章重點不同

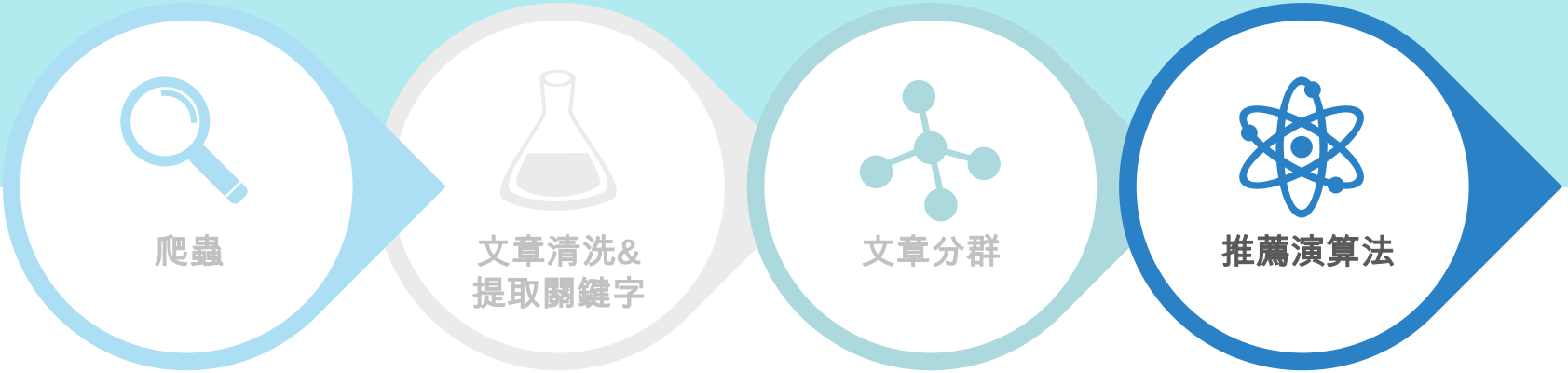
Group 1	Group 19	Group 27
減重	瘦身	減重
荷爾蒙	減重	維生素C
生活習慣	減肥	番茄
更年期	燃脂	排毒
飲食習慣	脂肪	蘋果
女性荷爾蒙	排毒	免疫力
自律神經	瘦身計畫	水果
失眠	高血壓	蔬菜
飲食	運動	蛋白質
呼吸	健康瘦	香蕉

- 分群結果符合一般常識

Group 2	Group 4	Group 11
高血壓	暈眩	便祕
心肌梗塞	肩頸僵硬	腸胃健康
心血管	耳鳴	腸道
腦中風	頭暈	腸胃
中風	脖子	減重
血管	頸椎	腸胃蠕動
心臟病	自律神經	排便
腦梗塞	淋巴	膳食纖維
動脈硬化	按摩	大腸癌

Group 19	Group 25	Group 21
膽固醇	血糖	失智症
脂肪肝	糖尿病	失智
動脈硬化	降血糖	大腦
高血脂	高血糖	預防失智
心血管	胰島素	阿茲海默症
心肌梗塞	高血壓	記憶力

Line bot-文章推薦

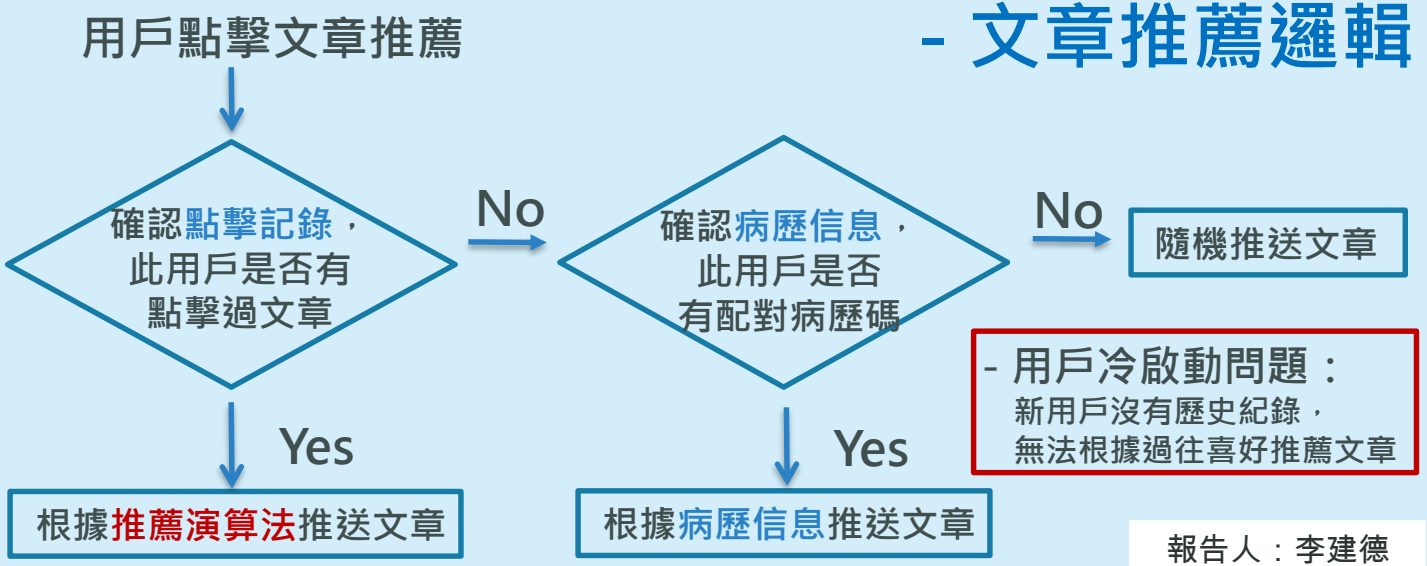


報告人：李建德

Line bot 文章推薦

- 收取點擊記錄

- 文章推薦邏輯



報告人：李建德

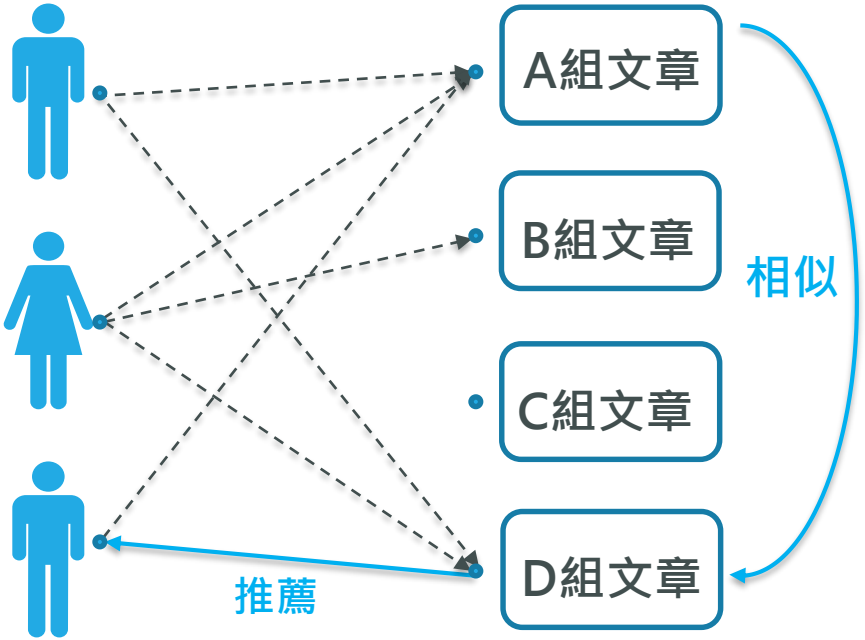
點擊記錄

lineID	time	URL
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-02 06:27:56	https://www.everydayhealth.com.tw//article/14470
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-02 06:30:38	https://www.everydayhealth.com.tw//article/17513
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-02 06:36:03	https://www.everydayhealth.com.tw//article/58
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-02 06:37:22	https://www.everydayhealth.com.tw//article/12648
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-02 06:39:27	https://www.everydayhealth.com.tw//article/20922
U83dc8574c8088af5d3160ce9570720d2	2019-09-02 09:00:28	https://www.everydayhealth.com.tw//article/10535
U83dc8574c8088af5d3160ce9570720d2	2019-09-02 09:00:55	https://www.everydayhealth.com.tw//article/11304
U38aa5aff206035fa1299abb1627db562	2019-09-07 08:45:01	https://www.everydayhealth.com.tw//article/17380
U52aff2b1837087007c2426f323ee43fa	2019-09-07 08:45:07	https://www.everydayhealth.com.tw//article/5620
U52aff2b1837087007c2426f323ee43fa	2019-09-07 08:45:15	https://www.everydayhealth.com.tw//article/16503

病歷信息

lineID	PNO	AGE	SEX	DEPTNAME	POSkey
1	11111111	66	女	骨科	腰椎 背部
1	11111111	66	女	胃腸肝膽科	肝 腹部
1	22222222	82	女	骨科	腰椎 背部
1	22222222	82	女	胃腸肝膽科	肝 腹部
1	22222222	82	女	神經外科	腰椎 背部
Ubc2d78f9e714ca16172ef2e74326aa16	99999999	63	女	骨科	腓骨 下肢
Ubc2d78f9e714ca16172ef2e74326aa16	99999999	63	女	一般外科	腦 頭 科
Ubc2d78f9e714ca16172ef2e74326aa16	99999999	63	女	骨科	膝 下肢

推薦系統 協同過濾 (基於文章)



- 系統冷啟動問題：
  系統剛上線沒有用戶紀錄
  無法創造文章關聯性

- 請同學及朋友試用，
  創造初始log -- 161筆
- 根據關鍵字尋找文章

- 點擊記錄：計算加權分數

lineID	time	group_n
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-18 10:31:03	12
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-18 10:28:30	22
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-18 09:18:58	27
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-15 14:52:43	27
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-15 06:38:11	5
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-15 06:38:03	16
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-14 06:09:55	16
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-14 06:09:35	3
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-14 06:09:09	20
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-14 05:32:20	27
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-14 05:32:13	27
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-12 07:00:13	27
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-10 13:54:01	27
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-10 13:53:45	27

最新3篇*1.0
次3篇 *0.5
再次3篇*0.2

```python
def weighting(user_dict:dict) -> list:
    """從log算出分數, 並編排成預處理的格式"""
    result_list = []
    for key in user_dict.keys():
    # 按照點擊文章時間加權(ex: 最新3篇*1, 次新3篇*0.5, 再次三篇*0.2)
    # 相同用戶依照文章tag加總
        wt_dict = {}
        for i in range(len(user_dict[key])):
            k = user_dict[key][i][0]
            v = (1 if i in [0,1,2] else 0.5
                 if i in [3,4,5] else 0.2 )
            wt_dict[k] = v if k not in wt_dict else wt_dict[k]+v
            if i == 8: break
        # print(f"user {key}: ",wt_dict)
        for kk in wt_dict.keys():
            result_list.append((key, (kk, wt_dict[kk])))
    return result_list
```

報告人：李建德

推薦系統 協同過濾 (基於文章)

- Key Value 資料處理

- 點擊記錄
  (lineID, time, group_n)

lineID	time	group_n
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-02 06:27:56	2
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-02 06:30:38	12
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-02 06:36:03	6
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-02 06:37:22	5
Ubc2d78f9e714ca16172ef2e74326aa16	2019-09-02 06:39:27	23
U83dc8574c8088af5d3160ce9570720d2	2019-09-02 09:00:28	14
U83dc8574c8088af5d3160ce9570720d2	2019-09-02 09:00:55	8
U38aa5aff206035fa1299abb1627db562	2019-09-07 08:45:01	27
U52aff2b1837087007c2426f323ee43fa	2019-09-07 08:45:07	3
U52aff2b1837087007c2426f323ee43fa	2019-09-07 08:45:15	25

⇩ 計算加權分數

- Key：lineID
- Value：(文章分組, 加權分數)
  (lineID, (group_n, score))

id	group & score
U0bc32241c72e08e1f1737c0d85c01002	(10, 4.9)
U0bc32241c72e08e1f1737c0d85c01002	(4, 0.2)
U13678ba0afcfa77283794986cb322d2b	(27, 2)
U2ab453d4be9065ca806d3880d2491b65	(2, 1)
U2ab453d4be9065ca806d3880d2491b65	(17, 1)

計算相關係數 ⇨

- Key：(文章A組,文章B組)
- Value ：(AB的相關係數, AB同時點擊次數)
  ((group_A, group_B), (rab, pairs))

```
((0, 2), (0.8320502943378436, 2))
((0, 5), (1.0, 1))
((0, 15), (0.9486832980505138, 2))
((0, 16), (1.0, 1))
((0, 17), (1.0, 1))
((0, 18), (0.9999999999999998, 2))
((0, 20), (0.8320502943378436, 2))
((0, 21), (0.9863939238321436, 2))
((0, 27), (0.9999999999999999, 2))
((1, 14), (1.0, 1))
((1, 18), (1.0, 1))
((1, 19), (1.0, 1))
((1, 21), (1.0, 1))
((2, 3), (1.0, 1))
((2, 5), (1.0, 1))
((2, 10), (1.0, 1))
((2, 15), (0.6139406135149204, 2))
((2, 16), (0.9191450300180576, 2))
((2, 17), (1.0, 1))
((2, 18), (0.8511025427814364, 3))
((2, 20), (0.38461538461538464, 2))
((2, 21), (0.7295372041400852, 2))
((2, 23), (1.0, 1))
((2, 24), (1.0, 1))
((2, 27), (0.599877588492
```

報告人：李建德

推薦系統 篩選與結果

篩選「相關係數」
與「被評分次數」

```python
# 設定條件, 搜尋相似文章
def search_similar(group, tag_pair_with_scores_list, s_threshold=0.85, a_threshold=1):
    similarity_threshold = s_threshold
    appearence_threshold = a_threshold
    tag = group
    result = list(filter(lambda x: (x[0][0] == tag or x[0][1] == tag)
                           and x[1][0] > similarity_threshold
                           and x[1][1] > appearence_threshold,
                           tag_pair_with_scores_list))
    if len(result) != 0:
        return tuple(i[0][1] for i in result)
    else:
        # print("Similar tags(article type) not found!")
        return None
```

((group_A, group_B), (rab, pairs))

```
((0, 2), (0.8320502943378436, 2))
((0, 5), (1.0, 1))
((0, 15), (0.9486832980505138, 2))
((0, 16), (1.0, 1))
((0, 17), (1.0, 1))
((0, 18), (0.9999999999999998, 2))
((0, 20), (0.8320502943378436, 2))
((0, 21), (0.9863939238321436, 2))
((0, 27), (0.9999999999999999, 2))
((1, 14), (1.0, 1))
((1, 18), (1.0, 1))
((1, 19), (1.0, 1))
((1, 21), (1.0, 1))
((2, 3), (1.0, 1))
((2, 5), (1.0, 1))
((2, 10), (1.0, 1))
((2, 15), (0.6139406135149204, 2))
((2, 16), (0.9191450300180576, 2))
((2, 17), (1.0, 1))
((2, 18), (0.8511025427814364, 3))
((2, 20), (0.38461538461538464, 2))
((2, 21), (0.7295372041400852, 2))
((2, 23), (1.0, 1))
((2, 24), (1.0, 1))
((2, 27), (0.599877588492
```

報告人：李建德

推薦系統 篩選與結果

篩選「相關係數」與「被評分次數」

根據上次點擊與條件，推薦四篇文章

另加上兩篇隨機文章，提供「非推薦」的選擇

keywords	group
健康飲食 糖 高血糖 高血壓 高血脂 心臟病	2
中醫養生 血管 促進血液循環 腦中風 高血壓/糖尿病飲食 降血壓	2
養生保健 泡澡 大腦 抗老 血液循環 呼吸	18
健康飲食 骨質疏鬆 牛奶 大豆異黃酮 減重	16
頭痛/偏頭痛 偏頭痛 聲音 噪音 血管 頭痛 腦神經	10
健康飲食 葉酸 維生素B群 糖尿病 降血糖	25

上次點擊第2組文章

與第2組相關的16, 18組

隨機文章

報告人：李建德

推薦系統 實際效果

點擊

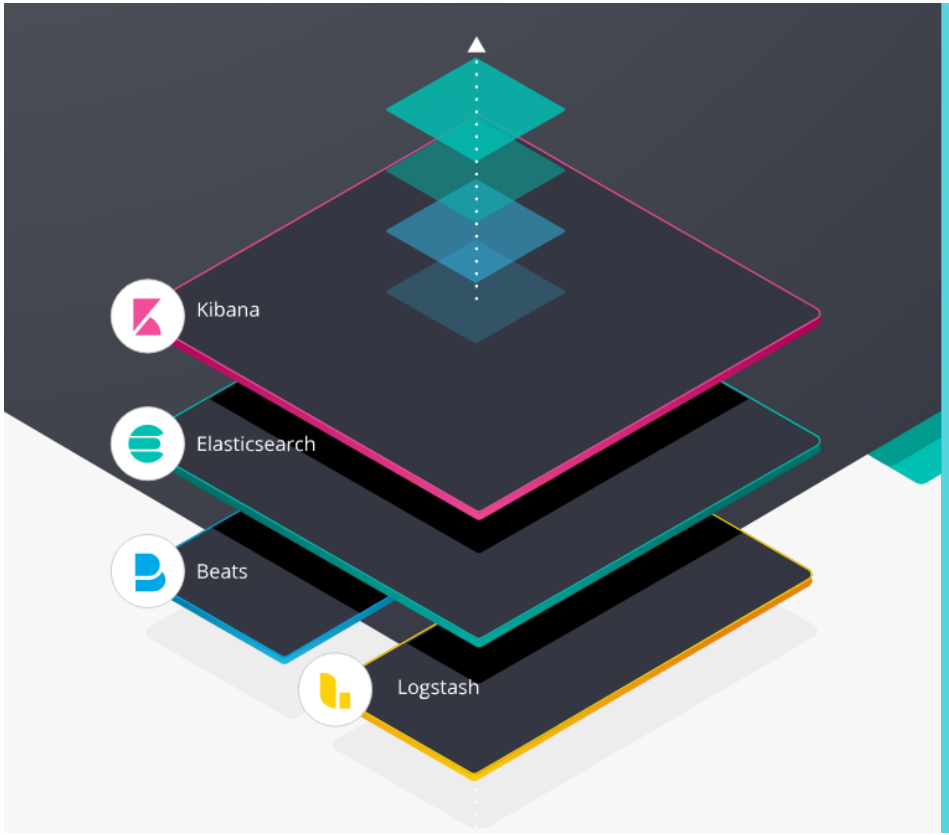
推薦

報告人：李建德

資料監控

報告人：賀俊賢


MEDICAL
Health Care
Doctor
Hospital
Pharmacist
Nurse
Dentist
First Aid
Surgeon
Emergency
MEDICAL
MEDICAL
MEDICAL



Elastic stack

Elasticsearch
儲存數據、
搜尋、分析

Kibana
匯總、分析、
視覺化呈現

Logstash
搜集、過濾、
傳輸

Beats
輕量型單一數
據採集器

報告人：賀俊賢

Why Elastic stack

1. 支援抓取種類多，包含文件、資料庫、系統log...
2. 即時索引功能，處理方式靈活不需預先編程
3. 檢索性能高效，可以達到秒級響應
4. **使用的JSON格式**，大部分程式語言都有支援讀寫
5. 前端操作簡單，多種圖表提供視覺化呈現

報告人：賀俊賢

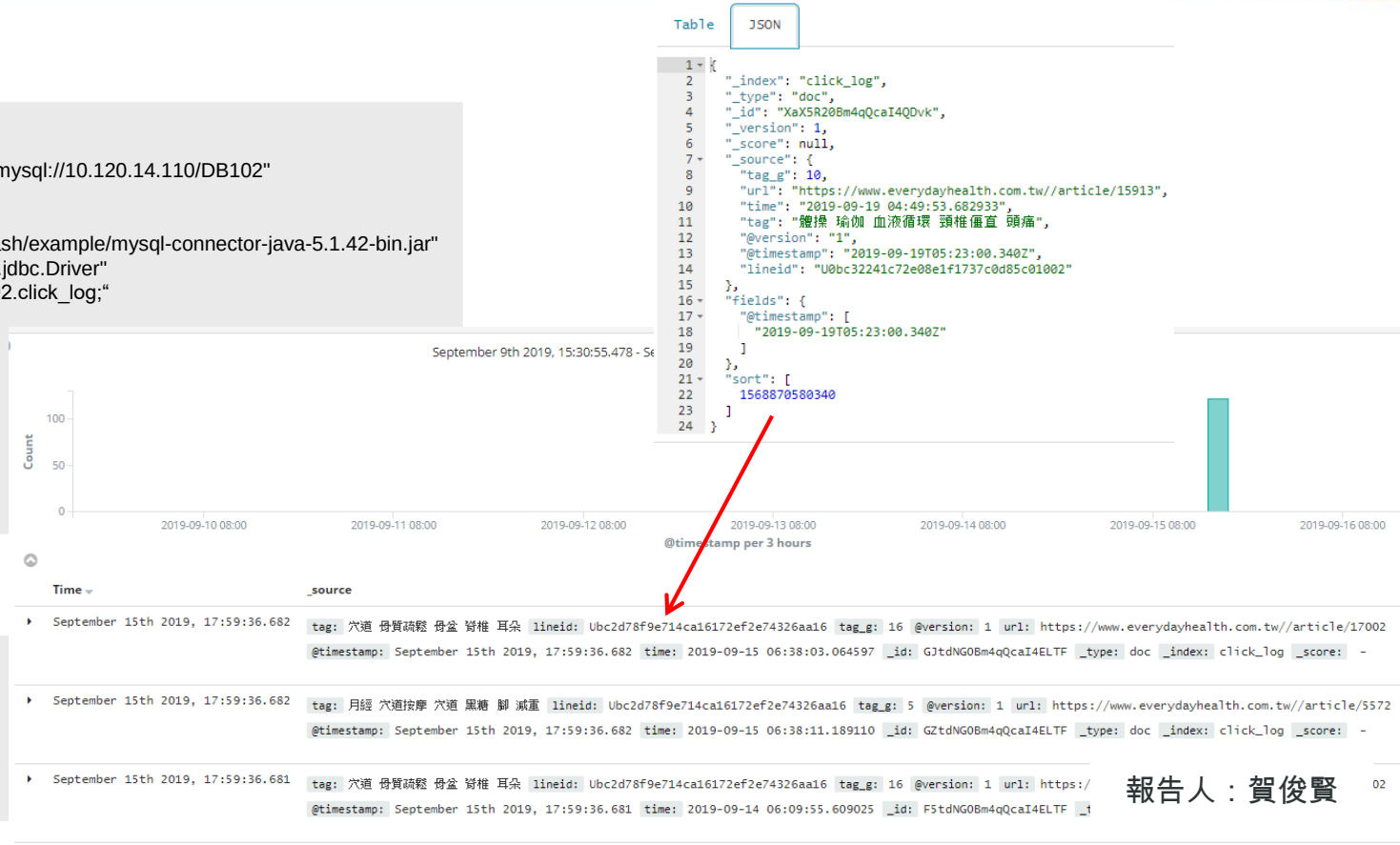
Logstash

mysql

```
input{
	jdbc{
		jdbc_connection_string => "jdbc:mysql://10.120.14.110/DB102"
		jdbc_user => "user"
		jdbc_password => "password"
		jdbc_driver_library => "/etc/logstash/example/mysql-connector-java-5.1.42-bin.jar"
		jdbc_driver_class => "com.mysql.jdbc.Driver"
		statement => "select * from DB102.click_log;“
		schedule => “* * * * *"
	}
output{
	elasticsearch{
		index => "click_log"
		hosts => ["localhost:9200"]
	}
	stdout {codec => json}
}
```

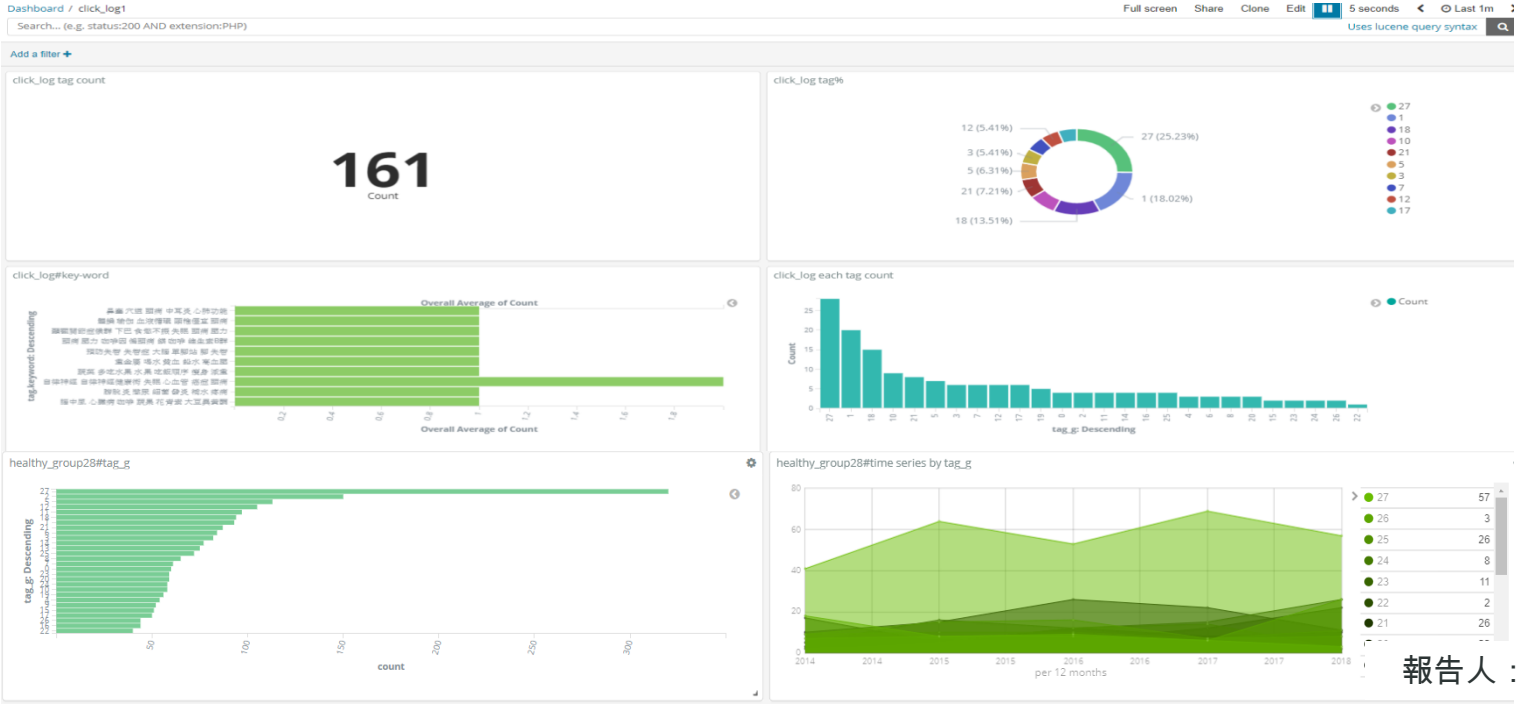
file

```
input{
	file{
		type => "log"
		path => ["/var/log/mysqld.log"]
		start_position => "beginning"
	}
}
```



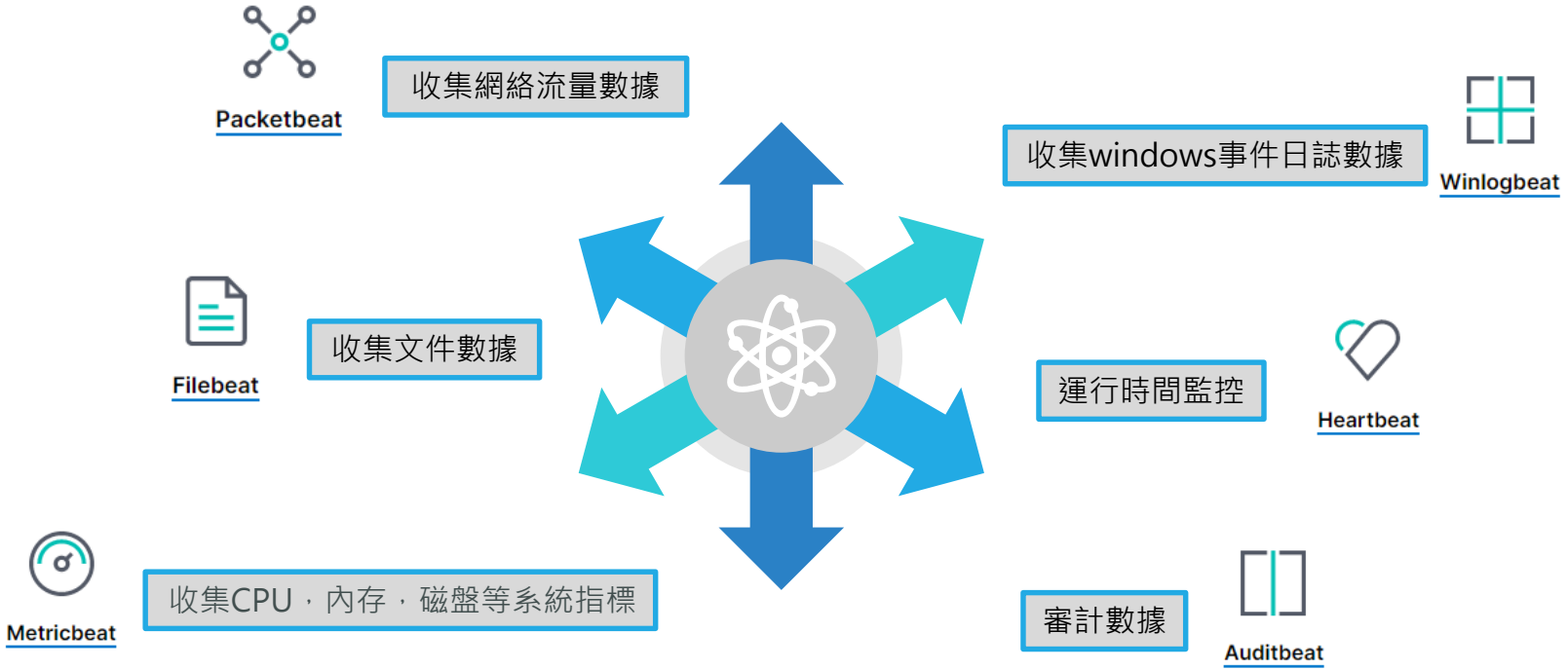
Kibana

Web呈現Elasticsearch儲存的資料，
選擇合適的圖表形式，
匯總於dashboard呈現。



報告人：賀俊賢

Beats

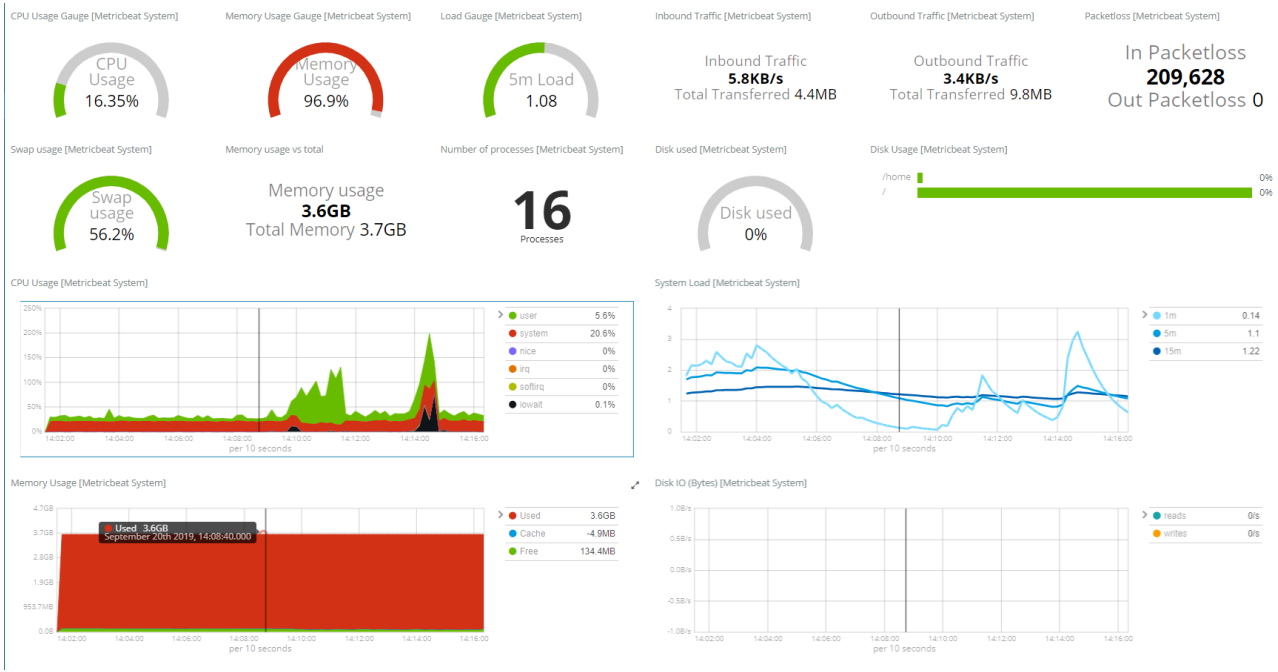


報告人：賀俊賢

Metricbeat

LogoType™

輕量級的系統級性能指標監控工具，
收集CPU，內存，磁盤等系統指標。




```
- module: system
  period: 10s
  metricsets:
    - cpu
    - load
    - memory
    - network
    - process
    - process_summary
    #- core
    #- diskio
    #- socket
  processes: ['.*']
  process.include_top_n:
    by_cpu: 5      # include top 5 processes by CPU
    by_memory: 5   # include top 5 processes by memory

- module: system
  period: 1m
  metricsets:
    - filesystem
    - fsstat
  processors:
  - drop_event.when.regexp:
      system.filesystem.mount_point: '^/(sys|cgroup|proc|dev|etc|host|lib)($|/)'

- module: system
  period: 15m
  metricsets:
    - uptime
```

報告人：賀俊賢

未來展望

1. 精密儀器檢查自動排程系統
2. 搭建混合雲雲端資料儲存
3. Line-Chatbot預約系統、掛號系統
4. Line-Chatbot自動推播系統

報告人：陳亞生


Medical Billing Statement
Admission Date : August 14, 2016
Discharge date
Service Date
8-14-16

謝謝聆聽

報告人：陳亞生