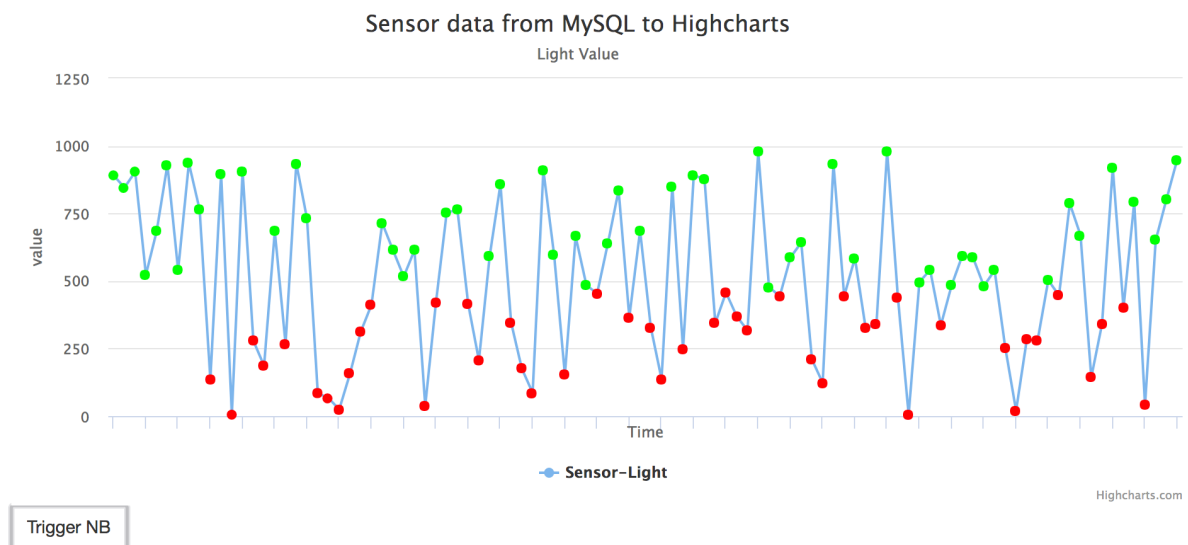


物聯網應用與資料分析 Assignment3 - 構建一個有專家系統的 IoT Web

姓名：吳嘉偉 學號：5105056013 日期：2018/4/28

1 目標

希望可以透過專家系統，把數值訓練成在某一數值以上或以下用不同的顏色區分，並把結果畫出曲線產生在網頁上。



2 產生資料庫資料

2.1 建立資料庫

利用 create.php 建立一個新的資料庫

程式碼：

```
<?php
/*
 * 自動建立資料庫及資料表
 * 欄位：
 *      id          -> int (主鍵)
 *      data        -> int
 *      time        -> TIMESTAMP (CURRENT_TIMESTAMP)
 */
//IP、帳號資訊
$host = "172.32.16.42";
$user = "iot";
$pass = "iot";
//資料庫資訊
$databaseName = "lightdb";
$tableName = "light";
//連線資料庫伺服器
$con = mysqli_connect($host,$user,$pass);
//建立資料庫
$sql = "CREATE DATABASE $databaseName";
mysqli_query($con, $sql);
//連結資料庫
$db = mysqli_select_db($con, $databaseName);
//建立資料表
$sql = "CREATE TABLE ".
    "$tableName"(
        'id' INT NOT NULL AUTO_INCREMENT,
        'value' INT NOT NULL,
        'status' INT NOT NULL,
        'time' TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
        PRIMARY KEY ('id')) ENGINE=InnoDB";
```

```
mysqli_query($con, $sql);  
?>
```

2.2 產生隨機的數值與狀態

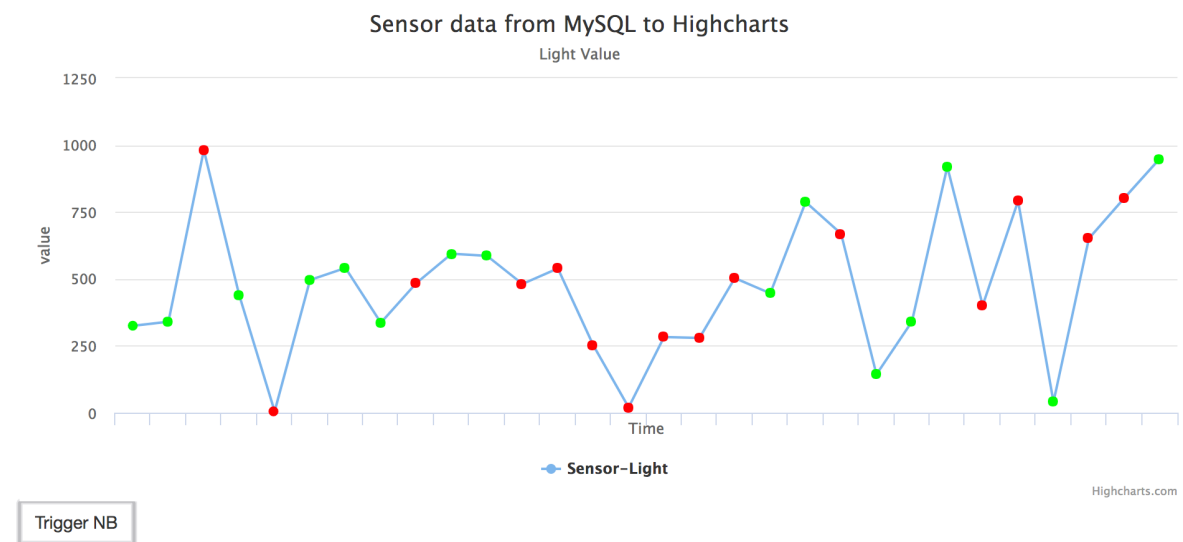
利用 dataset.php 隨機得產生數值與狀態

程式碼：

```
<?php  
  
$host = "172.32.16.42";  
$user = "iot";  
$pass = "iot";  
  
// 資料庫資訊  
$databaseName = "lightdb";  
$tableName = "light";  
  
// 連結資料庫  
$con = mysqli_connect($host, $user, $pass);  
$db = mysqli_select_db($con, $databaseName);  
  
for($i=0;$i<100;$i++) {  
    $sql = "INSERT INTO $tableName (value, status)  
          VALUES (".rand(0,1023).", ".rand(0,1).")";  
    mysqli_query($con, $sql);  
}  
  
?>
```

2.3 趨勢圖

到目前為止，我們就可以產生一個狀態是隨機的趨勢圖



3 專家系統

3.1 產生訓練資料

先用 Excel 產生一系列隨機亂數，並在第二列寫上狀態，如果數值大於 500 為 1，否則為 0，用此檔案當成訓練資料。

training

value	status
682	1
390	0
649	1
509	1
522	1
391	0
274	0
258	0
448	0
954	1
750	1
245	0
855	1

3.2 訓練資料

利用 Python 的 sklearn，把上一步驟產生的檔案用來訓練資料庫的資料並給予正確的狀態

```
# read training data
data = pandas.read_csv("training.csv")
pdX = data["value"]
npX = numpy.array(pdX)
npX = npX.reshape(-1, 1)

pdY = data["status"]
npY = numpy.array(pdY)

# sklearn
clf = lm.LogisticRegression()
clf.fit(npX, npY)
```

3.3 訓練資料庫數值

利用剛剛產生的 LogisticRegression 來訓練資料庫的資料，並把訓練完的結果寫入資料庫

```
# connect db
db = pymysql.connect("172.32.19.7", "iot", "iot", "lightdb")
cursor = db.cursor()

# 執行SQL語法查詢
cursor.execute("SELECT * FROM light")

# 整理搜尋出來的資料
id_list = []
value_list = []
results = cursor.fetchall()
```

```

for row in results:
    id_list.append(row[0])
    value_list.append(row[1])

# 把結果轉為ndarray
predictValues = numpy.array(value_list)
predictValues = predictValues.reshape(-1, 1)

# 把values丟到LogisticRegression預測狀態
predictStatus = clf.predict(predictValues)
status_list = predictStatus.tolist()

# 更新資料庫狀態
for i in range(len(id_list)):
    id = id_list[i]
    status = status_list[i]
    cursor.execute("update light_set status = %d
    where id = %d" % (status, id))
    db.commit()

# 關閉資料庫
db.close()

```

3.4 輸出結果

把訓練後的資料組成json格式後輸出回去

```

# 輸出結果
result = json.dumps([{'value': rowValue, 'status': status}
    for rowValue, status in zip(value_list, status_list)])
print(result)

```

4 HighChart

4.1 新增按鈕

在趨勢圖的下方新增一個按鈕，用來觸發訓練事件

```
<div class="container">
    <div id="container"
        style="min-width: 310px;
        height: 400px;
        margin: 0
        auto">Insert Highchart Here</div>
    <button class = "btn_byn-info"
        id = "trigger">Trigger NB</button>
</div>
```

```
$('#trigger').click(function(){
    $.ajax({
        url: 'triggerNB.php', // 連接的URL
        data: "{}", // 夾帶的參數
        dataType: 'json', // 資料格式
        success: function(data) // 傳送成功的function
        {
            alert("success");
        },
        error: function(XMLHttpRequest,
            textStatus, errorThrown) {
            alert("Error: " + errorThrown);
        }
    });
});
```


4.2 Php 呼叫 python

利用 `execu()`，呼叫 python，並把結果輸出給 highchart

```
<?php
    $output = exec( '/Library/Frameworks/Python.framework
    ...../Versions/3.6/bin/python3.6 ./ linear_model.py ');
    echo ($output);
?>
```

4.3 解析回傳資料

把 php 吐回來的資料解析後重新畫趨勢圖

```
values = [];
time = [];
for (var i = 0; i < data.length; i++)
{
    if(parseInt(data[i]['status']) == 0)
        values.push({y:parseInt(data[i]['value']),
            color: '#FF0000' });
    else
        values.push({y:parseInt(data[i]['value']),
            color: '#00FF00' });
    time.push(data[i]['time']);
}
highcharsinit();
```

4.4 執行結果

訓練後的結果，當 value = 480 左右做個區隔

