Programming assignment report

一、資料轉換

先載入 XML 檔，並讀取 XML 檔中的<Content>節點(數據所在的位置)，數據提取出來後將科學記號的部份去掉並轉為浮點數形式，接著透過迴圈先建立分類資料集的 list，如果在分類資料集中的有效值為 1，則此筆數據的實際溫度值也會同時被送到迴歸資料集的 list 當中，最後分別匯出兩個 csv 檔(classification.csv 與 regression.csv)。

二、模型訓練

（一）、分類模型

一開始使用 Logistic regression，但是遇到精確度不足以及訓練時間過長的問題，因此改用 random forest 來訓練，random forest 的訓練方式為一開始一樣將資料及分成 80%與 20%的 test set 與 validation set，接著會建立 100 個集合(稱為樹)，每棵樹裡都有 120*67*0.8 個資料點，而這些資料點是由原先的 csv 檔中隨機取的，且取的方式為取後放回，所以同一個數據可能被取到多次。

接著每棵樹會進行分層的分類，分類的依據(稱為特徵)會隨機選擇經度或緯度，接著根據特徵的數值進行大小二分法(此數值稱為分裂點)，例如緯度小於 120.5 的一組，大於等於的為另一組，在同一層中會先試過各種不同的分裂點，最後找到純度最高的，也就是是否為有效值 1,0 分的最乾淨的一種分法，再從這樣的分法下，繼續下一層的分類，每一層都以此方法一直分類下去直到純度足夠高或是不能再分支下去為止。

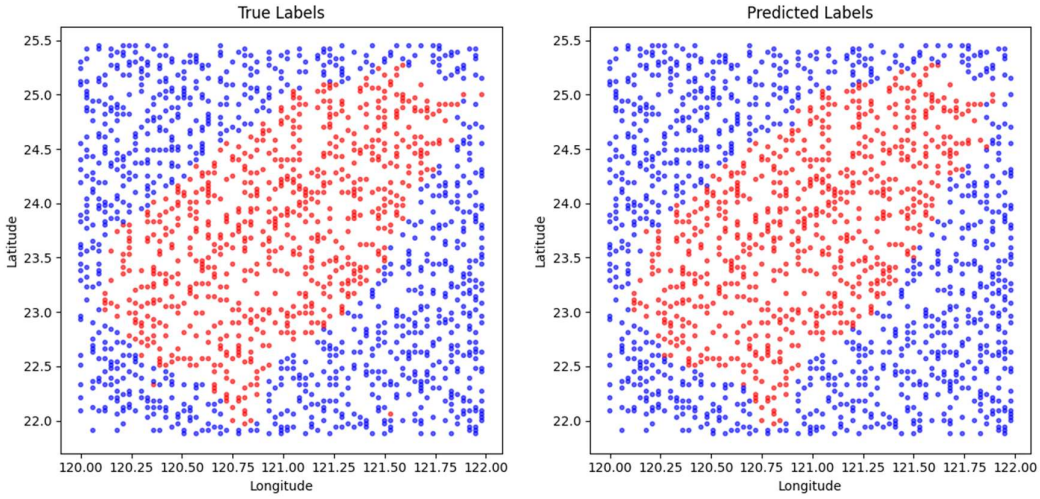
所以依照上面的步驟，給定一個想要預測的經緯度時，會根據 100 棵樹自己的分類方式找的 100 種預測值，接著選出出現次數較多的預測值當作最終預測值，舉例來說，有 80 棵樹預測值為 1，20 棵樹的預測值為 0，則最後採用 1 來當此點的預測值。

（二）、迴歸模型

迴規模型同樣使用 random forest 來訓練，與分類模型不一樣的是，每棵樹有的資料點數量是等同於 regression.csv 檔內的數量*0.8，另外在每棵樹的預測值同樣是透過不斷的二分法來決定，但是在最終預測值的部分是由 100 棵樹的預測值去做平均，而不是用多數決的方式產生。

三、訓練結果

（一）、分類模型

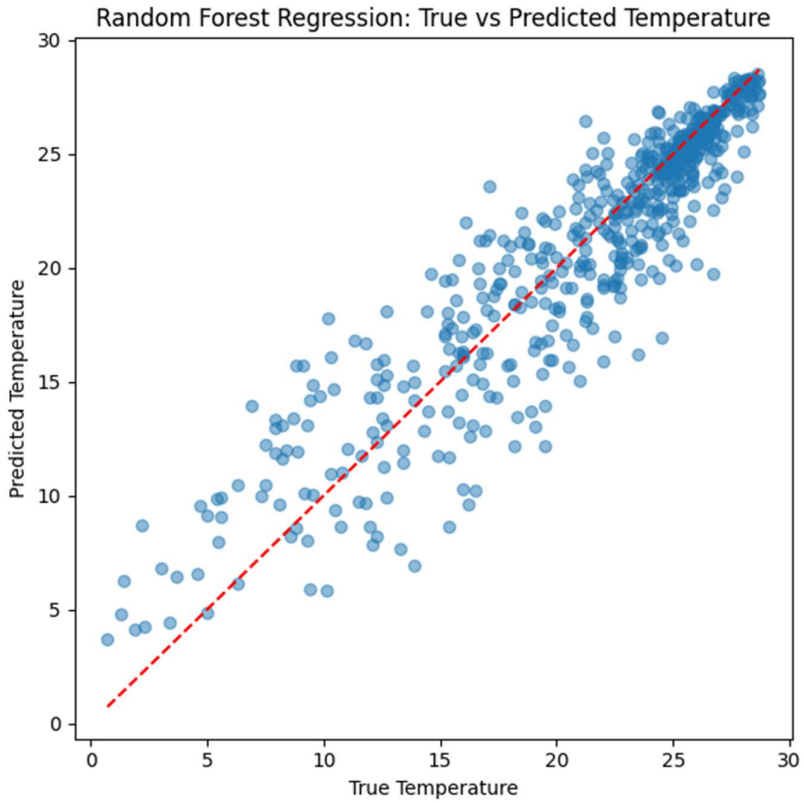


上圖是 validation set 驗證的結果，左圖實際值，右圖是預測值，紅色代表此經緯度為有效值，藍色則代表無效值

最終的精確度約為 0.98，此精確度是透過以下計算所得來：

$$\frac{validation\ set\ 中預測正確的點數量}{validation\ set\ 中所有的點數量}$$

（二）、迴歸模型



上圖縱座標代表預測的溫度，橫軸代表實際溫度，紅色虛線為輔助判斷，越靠近紅色虛線代表誤差值越低，但此圖看不出各點的經緯度，因此有另外輸出 regression_prediction_with_coordinates.csv 檔來將經緯度、預測值、實際值都放在一起

誤差值選用 MSE 與 max error 做計算：

```
Mean Squared Error (MSE): 4.819634965852808
Max Error: 7.623000000000295
```