Programming assignment

一、Classification using GDA

1. 如何使用 GDA:

一開始一樣先將所有的數據分成80%與20%的訓練集與預測集，接著令$x \in \mathbb{R}^2$，x的兩個分量分別代表經度與緯度，而y的可能值為0,1，透過 GDA 的模型假設當$y = 0\ or\ 1$時$x \sim N(\mu \in \mathbb{R}^2, \Sigma \in M_{2\times2})$，且這邊透過 LDA 的假設所以Σ是共用的，接著利用 week5 written assignment 中 MLE 的結果與訓練集的數據計算μ_0, μ_1, Σ，而$\phi_1\ or\ \phi_0$則是$y = 1\ or\ 0$在整體數據的個別占比，這5個參數可以決定出兩個高斯分佈，分別是經緯度對$y = 1\ or\ 0$的機率，以下為計算參數的程式碼。

```
# 計算mu, phi, sigma
def fit_gda(X, y):
    labels = np.unique(y)
    mu = np.array([X[y == l].mean(axis=0) for l in labels])
    phi = np.array([np.mean(y == l) for l in labels])
    sigma = sum([((X[y == l] - mu[i]).T @ (X[y == l] - mu[i])) for i, l in enumerate(labels)]) / len(X)
    return mu, phi, sigma, labels
```

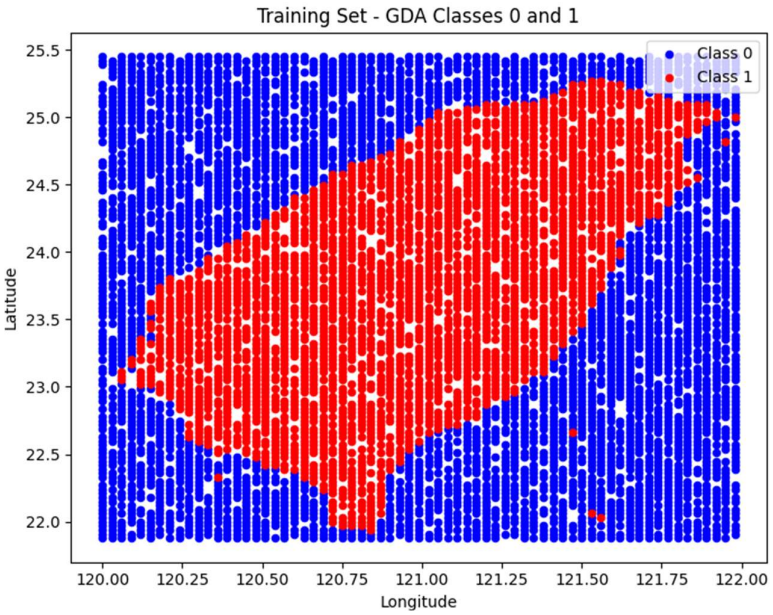
接著當決定完兩個高斯分布後可以開始進行預測，給定一個經緯度x_0，比較$P(x_0|y = 0)$、$P(x_0|y = 0)$的大小，選擇較大的作為預測值，以下為透過參數計算預測值的程式碼。

```
# 預測函式
def predict_gda(X, mu, phi, sigma, classes):
    inv_sigma = np.linalg.inv(sigma)
    probs = []
    for i, class_label in enumerate(classes):
        a = -0.5 * np.sum((X - mu[i]) @ inv_sigma * (X - mu[i]), axis=1)
        a += np.log(phi[i])
        probs.append(a)
    probs = np.vstack(probs)
    return classes[np.argmax(probs, axis=0)]
```

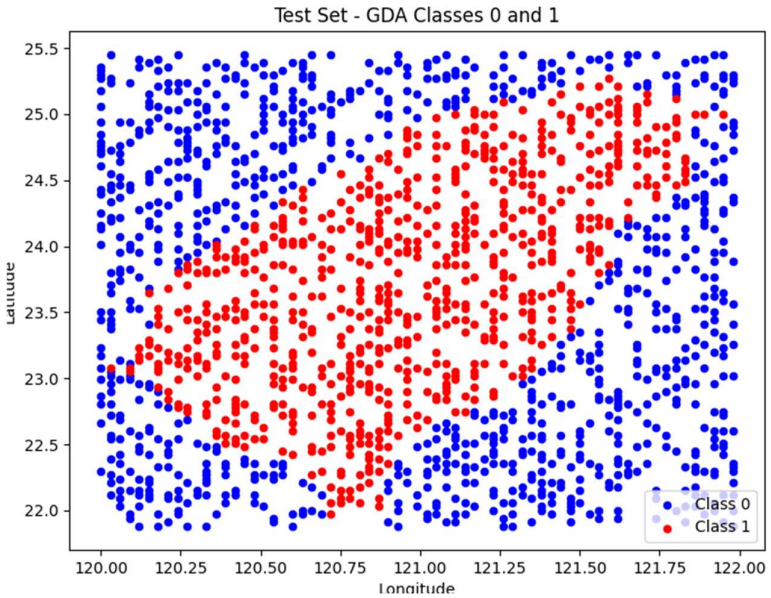
2. 為何可以使用 GDA

因為有效值的點從數據的來源來看，在台灣的土地上會是1，海上則會是0，所以二元分類的兩個類別的分佈是集中的，所以從高斯分布的等高線來看，台灣可以用一個橢圓形圈出來，而透過兩個類別的高斯分佈相互比較，可以更增加精確度。

3. 訓練結果



上圖為訓練集訓練後的結果



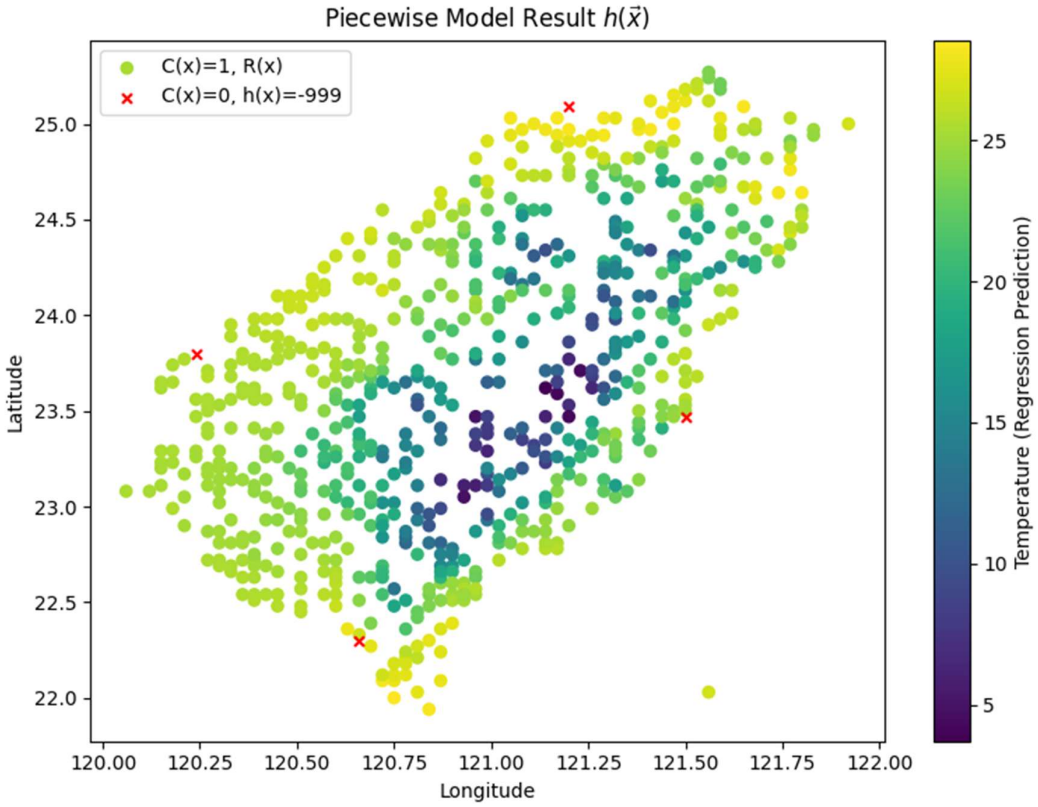
上圖為預測集測試的結果

最後的精確度透過預測集中正確預測的點數量÷所有點數量得到下列結果:

```
Test Accuracy: 0.5143
```

二、Regression

在前半段都與 assignment 4 的內容相同，也就是透過 random forest 來訓練 classification 和 regression 模型，接著先將兩組訓練出來的結果進行合併，並透過 numpy 中的 where 函數來判別要輸出的是−999或預測值，也就是$h(x)$，而在畫圖的部分透過判斷經緯度的有效值來判斷是否要將點畫出來，且顯示點的顏色依據溫度由低到高做漸層，結果如下圖:



如果要看詳細的輸出數據有另外將各點的$h(x)$預測值輸出到 combined_model_output,csv 檔中的第 5 欄。