迴歸學習器的介紹

周芊妤

January 9, 2024

在分析資料並預測未知資料時，我們需要掌握重要的技能，本文運用了三種迴歸學習器：簡單線性迴歸模型、增廣迴歸模型和邏輯斯迴歸模型，透過這些模型在散佈圖上描繪出了資料間的分界線，並進行更深入的資料預測。這些方法不僅幫助理解資料間的關聯，更提供了可靠的工具來預測未知資料，強化了對資料特性和未來趨勢的洞察力，這樣的分析和預測能力對於解決實際問題和做出準確的決策至關重要。

1 迴歸學習器

迴歸學習器是一種機器學習模型，它被用來預測連續值的輸出，當討論到迴歸學習器時，通常在探討一種學習算法，用於預測數值型目標（稱為因變量或應變量）的值，基於其他特徵（稱為解釋變量或自變量）的數值，這種方法是在建立一種模式來描述解釋變量和目標變量之間的關係。以下是三種常見的迴歸學習器：

1. 線性迴歸模型（Linear Regression Model）：
   線性迴歸模型所繪製的數據散點圖分界線的目的是擬合數據點的分佈，使其盡可能地代表整體數據趨勢，並在其附近最大化數據點的集中程度，這條直線通常是通過最小化觀測值和迴歸線之間的誤差來擬合的，誤差可以是垂直方向上的距離，也可能是其他一些衡量誤差的方法，此條線可用於預測數據、描述其趨勢，並為未來的數據點提供一種近似的預測。

2. 增廣迴歸模型（Augmented Regression Model）：
   除了簡單線性迴歸使用單個解釋變數外，擴充迴歸模型還使用了多個解釋變數（ $X_1,\ X_2,\ X_1X_2,\ X_1^2,\ X_2^2$ 以及 $X_3\cdots$ 等等），這種模型更加靈活，可以處理多元線性的關係。

3. 邏輯斯迴歸模型（Logistic Regression Model）：
   實際上是用於分類問題的一種迴歸模型，主要用於預測二元結果（0 或 1），例如

二元分類問題，使用邏輯函數將輸出轉變成在 0 和 1 之間，表示機率。

這些模型有不同的應用和適用情況，簡單線性迴歸模型有助於捕捉特徵間的線性關係，適用於預測特定數值及特徵的情況；增廣迴歸模型則拓展了線性迴歸，更靈活地處理非線性關係，適用於更複雜的多特徵問題；而邏輯斯回歸模型主要用於分類問題，能有效將數據分為不同的類別。

2 生成兩群資料

本節生成兩群資料，圖 1 中的兩筆資料分別是由藍色以及橘色兩群所生出，它們的資料內容是：

1. 藍色：設定樣本數為 200 ，中心點座標為 $(-3, 2)$ ，共變異矩陣（Covariance Matrix）為 $\begin{bmatrix} 10 & 1.2 \\ 1.2 & 2.3 \end{bmatrix}$ ，此矩陣描述了兩個變數之間的變異數和共變異數，對角線上的值代表每個變數自身的變異數（第一個變數的變異數為 10 ，第二個變數的變異數為 2.3 ），非對角線上的值表示兩個變數之間的共變異數（這裡為 1.2 ）。
2. 橘色：設定樣本數為 200 ，中心點座標為 $(3, 2)$ ，變異 - 共變異矩陣為 $\begin{bmatrix} 4 & -0.9 \\ -0.9 & 2 \end{bmatrix}$ 。

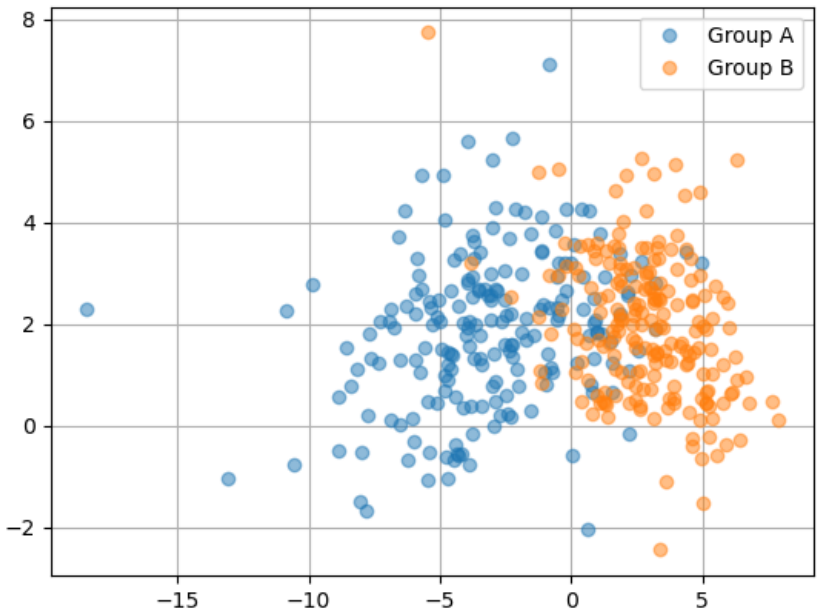
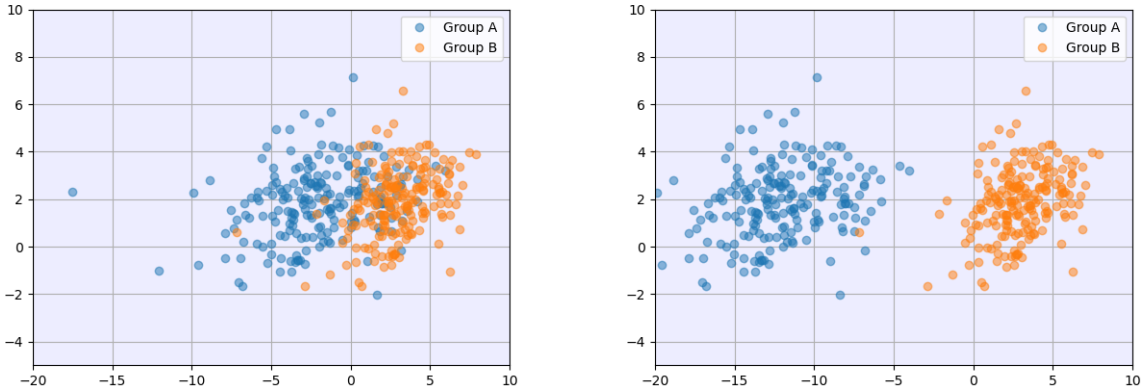


圖 1: 兩群資料散佈圖

接下來，試著調整中心點座標、變異數以及共變異數的數值，並分別從圖中看出彼此的差別。

2.1 調整中心點座標

固定共變異矩陣為藍 $\begin{bmatrix} 10 & 1.2 \\ 1.2 & 2.3 \end{bmatrix}$ 以及橘 $\begin{bmatrix} 4 & -0.9 \\ -0.9 & 2 \end{bmatrix}$ ，接著調整中心點座標，圖 2 (a) 中為藍中心點座標 $(-2, 2)$ 以及橘中心點座標 $(3, 2)$ ，圖 2 (b) 中為藍中心點座標 $(-12, 2)$ 以及橘中心點座標 $(3, 2)$ ，圖 2 中可以比較出當兩個座標分得越開時，兩群的距離亦隨之越遠。

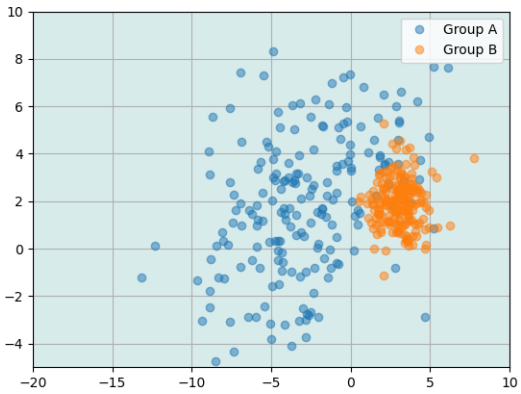


(a) 中心點座標藍 $(-2, 2)$ 以及橘 $(3, 2)$　　(b) 中心點座標藍 $(-12, 2)$ 以及橘 $(3, 2)$

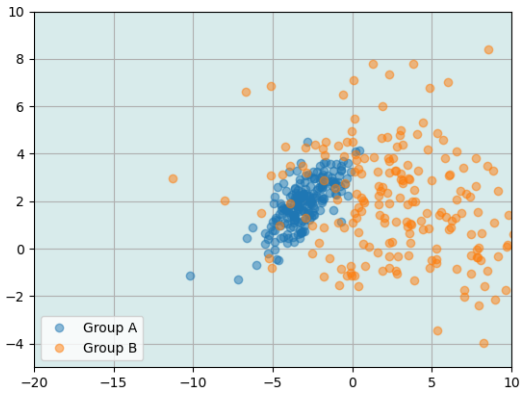
圖 2: 中心點座標位置不同的比較

2.2 調整變異數以及共變異數

固定中心點座標為藍 $(-3, 3)$ 以及橘 $(3, 2)$ ，調整變異數以及共變異數，圖 3 (a) 的變異 - 共變異矩陣為藍 $\begin{bmatrix} 15 & 5 \\ 5 & 10 \end{bmatrix}$ 以及橘 $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ ，圖 3 (b) 的變異 - 共變異矩陣為藍 $\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$ 以及橘 $\begin{bmatrix} 18 & -3 \\ -3 & 6 \end{bmatrix}$ ，圖 3 中可以看出當變異數越大時，點會越分散，反之，當變異數越小時，點會越集中；另外，當共變異數為正時，表示特徵之間呈現正相關，也就是說當一個特徵增加時另一個特徵也傾向於增加，反之，當共變異數為負時，表示特徵之間呈現負相關，也就是說，當一個特徵增加時另一個特徵傾向於減少。



(a) 藍色的變異數較大、橘色的變異數較小

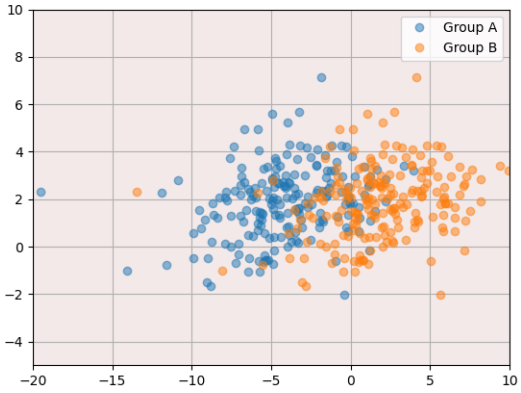


(b) 藍色的變異數較小、橘色的變異數較大

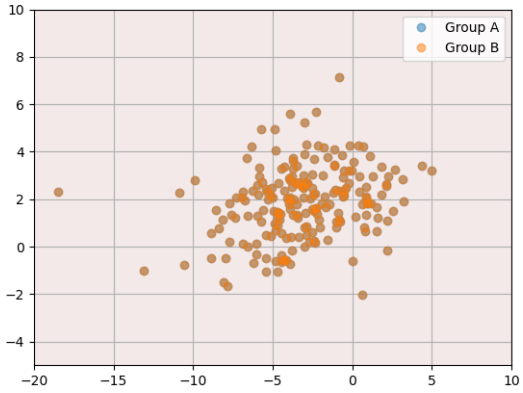
圖 3: 變異數以及共變異數不同的比較

2.3 在變異數以及共變異數相同的情況下調整中心點座標

試著將兩群資料的變異數以及共變異數設為一樣，去比較中心點座標不同的影響，圖 4 (a) 為中心點座標為藍 $(-4, 2)$ 以及橘 $(2, 2)$ ，可以看出兩群的散布狀況相同只是位置不同，圖 4 (b) 為中心點座標藍、橘同為 $(-3, 2)$ ，兩組數據的中心點相同，兩者的資料也完全重疊，顯示出中心點座標以及變異數以及共變異數的相同對於資料重疊度的重要性。



(a) 中心點座標為藍 $(-4, 2)$ 以及橘 $(2, 2)$



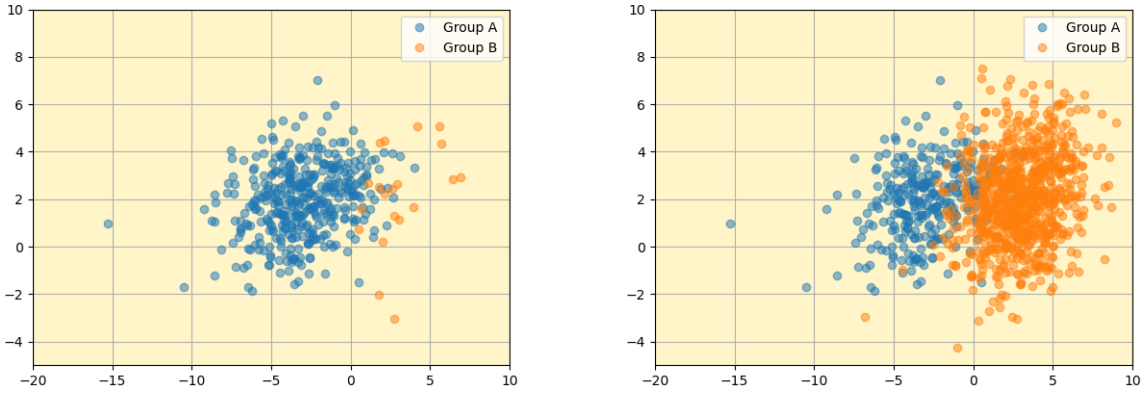
(b) 中心點座標藍、橘同為 $(-3, 2)$

圖 4: 變異數以及共變異數相同，中心點座標不同的比較

2.4 調整樣本數比較資料量之不同

試著調整樣本數的大小來比較資料量之不同，圖 5 (a) 中的樣本數為藍 400 以及橘 20 ，可以很明顯地看出藍色那群資料量較大也更加密集，反之，橘色那群資料量較少也更

加稀疏，圖 5 (b) 中的樣本數為藍 300 以及橘 800 ，可以看出兩群的資料量都很大也很密集，但藍色組相對於橘色組的資料量仍較少，也呈現出更加稀疏的特性。透過調整樣本量，能夠清晰展示資料量對於分布形態的影響，進而更深入理解資料密度與分佈間的關係。



(a) 樣本數為藍 400 以及橘 20　　(b) 樣本數為藍 300 以及橘 800

圖 5: 調整樣本數

3 資料散佈圖分界線

使用三種迴歸學習器來畫出圖 1 中資料散佈圖分界線。

1. linear regression model (線性迴歸模型)

   用於建模自變數 Y 和不同解釋變數 X 之間的線性關係，目標是找到一條最適合數據的直線，以最小化實際觀測值與模型預測值之間的差異。

   在圖 6 中，深紫色的直線是使用下列 Python 語法來做出由 Simple linear regression model 畫出的資料散佈圖分界線：

```
Mdl = LinearRegression()
Mdl.fit(X, y)
b = Mdl.coef_
intercept = Mdl.intercept_
x = np.array([x_min, x_max])
f =  -(intercept - 0.5 + b[0]*x) / b[1]
y_hat = Mdb.predict(X)
y_pre = [1 if i > 0.5 else 0 for i in y_hat]
logr=plt.plot(x, f, lw=1.5,label = 'Simple linear
   regression model'.format(1 - np.mean(y_pre == y)),color
   ='#484891')
```

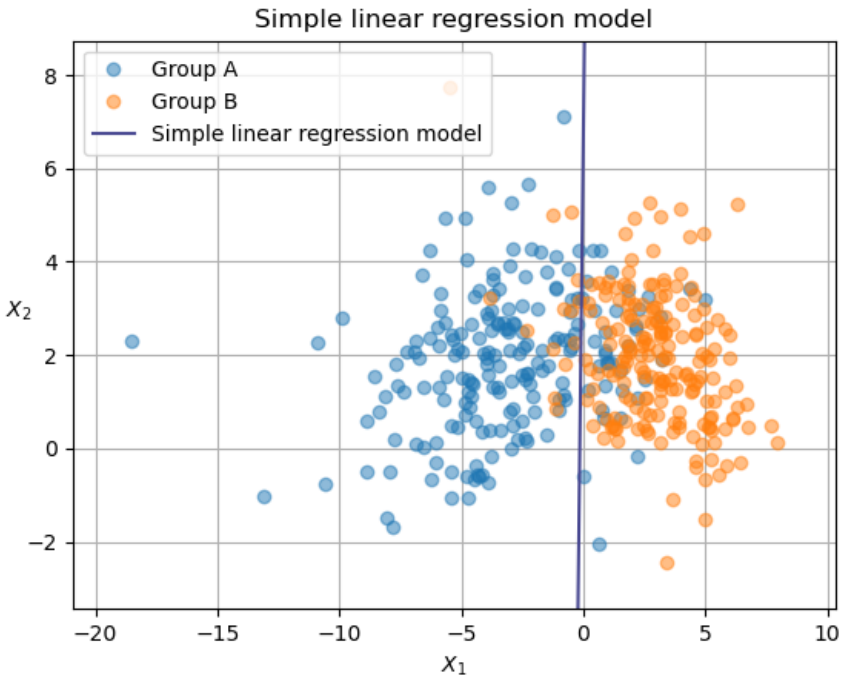


圖 6: Simple linear regression model

2. Augmented regression model (增廣迴歸模型)

增廣迴歸模型可以通過引入更多的特徵，進一步改進分類器的能力，這些特徵可以是原始資料的變換、多項式特徵、交互項或其他轉換後的特徵，透過這種方式，增廣迴歸模型可以增加模型的複雜度，使其能夠更好地擬合輸入資料的分佈，進而提高對新資料的預測準確性。

對於資料點的分類，繪製分界線的目的是找到一個決策邊界，它可以將資料空間分為不同的類別區域，對於線性分類器，這個決策邊界通常是一條直線（對於二維資料）或一個超平面（對於更高維度的資料），而此增廣迴歸模型決策邊界的位置是通過模型訓練來確定的，目的是最大化對訓練數據的準確性。

在圖 7 中，綠色的虛線是使用下列 Python 語法來做出由 Augmented regression model 畫出的資料散佈圖分界線：

```
Mda = LinearRegression()
x1, x2 = D[:, 0:1], D[:, 1:2]
X = np.hstack((x1, x2, x1*x2, x1**2, x2**2))
y = D[:, 2]
Mda.fit(X, y)
b = Mda.coef_
intercept = Mda.intercept_
f = (lambda x : intercept+ b[0]*x[0]+ b[1]*x[1]+ b[2]*x
   [0]*x[1]+ b[3]*x[0]**2+ b[4]*x[1]**2)
```

```
xx = np.linspace(x_min, x_max, nx)
yy = np.linspace(y_min, y_max, ny)
XX, YY = np.meshgrid(xx, yy)
Z = f([XX, YY])
Mda.fit(X, y)
y_hat = Mda.predict(X)
y_pre = [1 if i > 0.5 else 0 for i in y_hat]
contours = plt.contour(XX, YY, Z, levels = [0.5], colors
   =’#73BF00’, linestyles=’--’, label=r’Augmented
   Regression Model’)
```

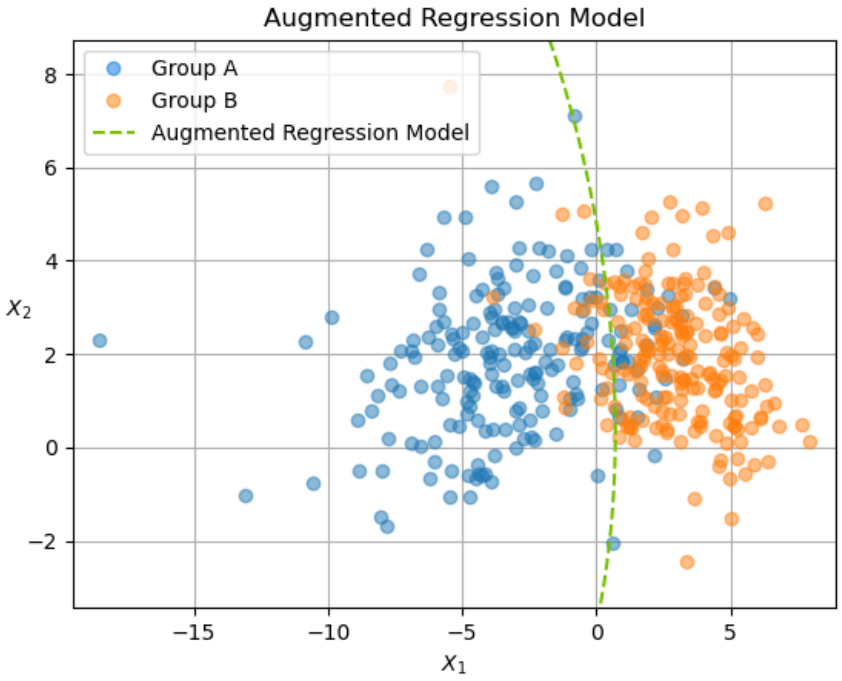


圖 7: Augmented regression model

3. Logistic Regression model (邏輯斯迴歸模型)

邏輯斯迴歸模型是一種廣泛用於二元分類問題的機器學習模型，它的原理是通過尋找一個能夠在資料空間中分隔不同類別的決策邊界，從而將資料點分類到相應的類別。

在繪製資料散佈圖的分界線時，邏吉斯迴歸模型使用 S 形曲線將輸入資料的線性組合映射到 0 到 1 之間的機率值，當機率值超過一個閾值（通常是 0.5 ）時，資料點被分類為一個類別，否則被分類為另一個類別。

邏輯斯迴歸通過最大概似函數來訓練模型，使得模型能夠找到最適合的決策邊界，從而最好地區分不同類別的資料點。訓練過程中，模型會更新權重以最大化正確分類的機率，進而找到最佳的分界線。

在圖 8 中，粉色的直線是使用下列 Python 語法來做出由 Logistic Regression model 畫出的資料散佈圖分界線：

```
Mdb = LogisticRegression()
X = D[:, 0:2]
Mdb.fit(X, y)
intercept = Mdb.intercept_
b = Mdb.coef_
x = np.array([x_min, x_max])
f = -(intercept + b[0,0]*x) / b[0,1]
y_hat = Mdb.predict(X)
y_pre = [1 if i > 0.5 else 0 for i in y_hat]
logr=plt.plot(x, f, lw=1.5, color='#D9006C',label = '{:.4
   f}'.format(1 - np.mean(y_pre == y)))
```

圖 8: Logistic Regression model

在圖 9 中是把簡單線性迴歸模型、增廣迴歸模型以及邏輯斯迴歸模型畫在同一張圖上，可以看出用這三種不同方法所生出來的資料散佈圖分界線會有差異。若想比較哪一個模型所生出的分界線最好的話，可以透過計算**訓練資料的誤判率以及準確率**來求得。

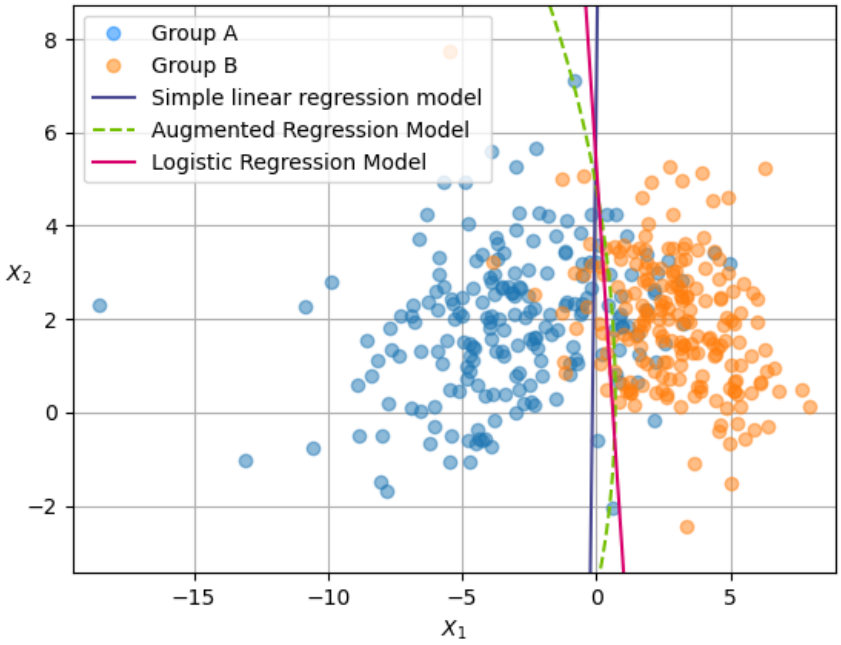


圖 9: All the models

4 訓練資料、測試資料的誤判率、準確率

在訓練機器學習模型時，會使用部分資料來訓練模型，再使用另一部分資料來測試模型的表現，當模型在訓練資料上進行預測時，如果預測結果與實際結果不符就會產生誤判，訓練資料的誤判率即是這些誤判的比率，而測試資料的誤判率是指模型在測試集上錯誤分類的比率，通常以百分比來表示。

誤判率可由以下公式計算：

$$\text{誤判率} = \frac{\text{錯誤預測的樣本數}}{\text{總樣本數}} \times 100\%$$

準確率可由以下公式計算：

$$\text{準確率} = \left(1 - \frac{\text{錯誤預測的樣本數}}{\text{總樣本數}}\right) \times 100\%$$

接著，試著做訓練資料、測試資料的誤判率、準確率，圖 10 為原始資料，是由兩群資料所生之散布圖，兩群資料內容如下：

1. 藍色：設定樣本數為 400 ，中心點座標為 $(-3, 2)$ ，變異 - 共變異矩陣（Variance-Covariance Matrix）為 $\begin{bmatrix} 10 & 1.2 \\ 1.2 & 2.3 \end{bmatrix}$ 。

2. 橘色 : 設定樣本數為 400 ，中心點座標為 $(3, 2)$ ，變異 - 共變異矩陣為 $\begin{bmatrix} 4 & -0.9 \\ -0.9 & 2 \end{bmatrix}$ 。

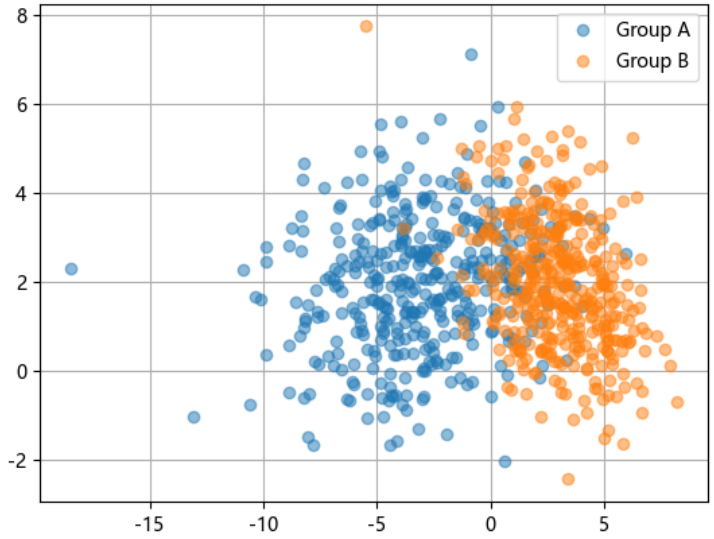


圖 10: 原始資料

圖 11 是把圖 10 中的樣本取 80% 做訓練資料、 20% 做測試資料，並用公式求出訓練資料、測試資料的準確率，上方為訓練資料，下方為測試資料，在簡單線性迴歸模型中，訓練資料的準確率為 88.28% 、測試資料的準確率為 88.12% ；在增廣迴歸模型中，訓練資料的準確率為 87.66% 、測試資料的準確率為 89.38% ；在邏輯斯迴歸模型中，訓練資料的準確率為 87.34% 、測試資料的準確率為 88.12% 。

透過**準確率越高迴歸模型越適合**的判斷可以得到 :

1. 訓練資料

   在訓練資料中，最適合此資料散布圖的迴歸模型為簡單線性迴歸模型。

2. 測試資料

   在測試中，最適合此資料散布圖的迴歸模型為增廣迴歸模型。

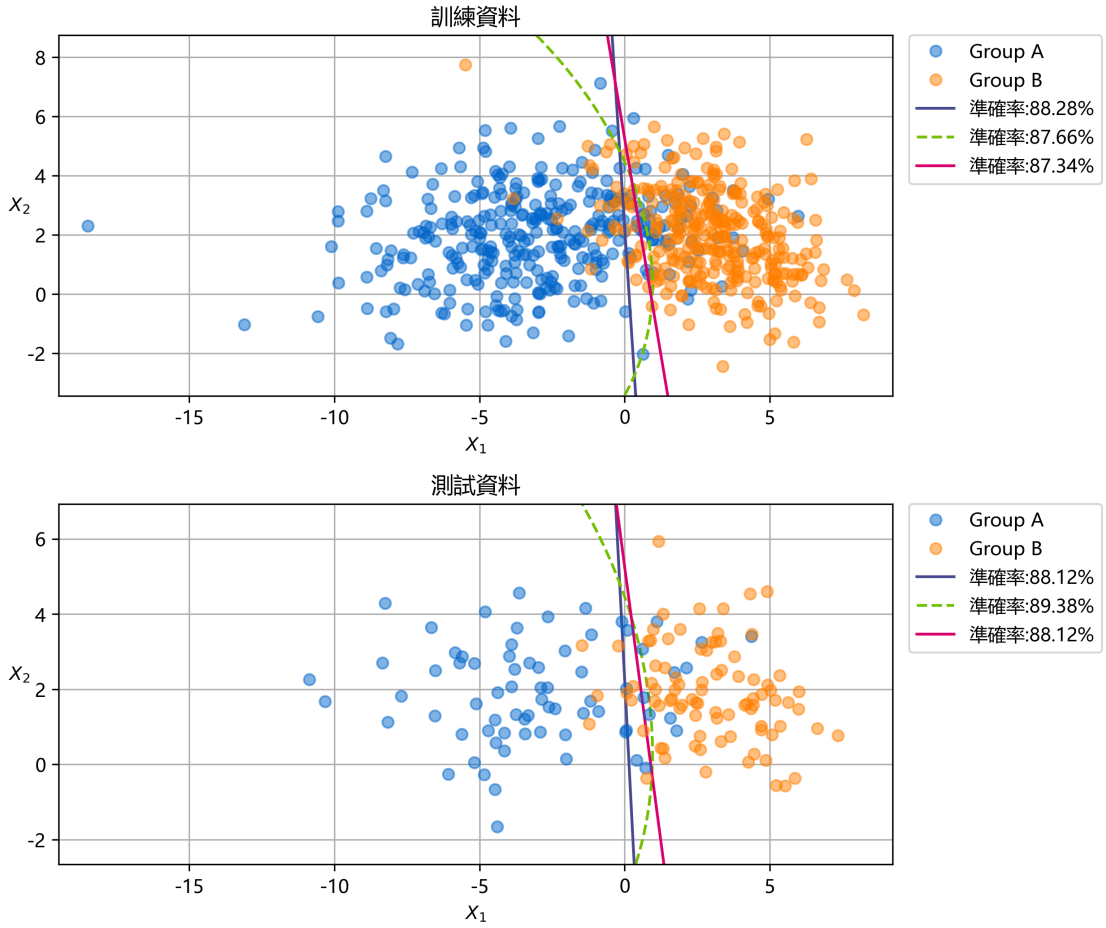


圖 11: 訓練資料、測試資料的準確率

5 生成三群資料

用 Python 來生成三群資料，圖 12 中的三筆資料分別是由藍色、橘色以及粉色三群所生出，它們的資料內容是：

1. 粉色：設定樣本數為 200 ，中心點座標為 $(0, -1)$ ，變異 - 共變異矩陣為 $\begin{bmatrix} 8 & 0 \\ 0 & 2 \end{bmatrix}$ 。
2. 綠色：設定樣本數為 200 ，中心點座標為 $(2, 4)$ ，變異 - 共變異矩陣為 $\begin{bmatrix} 3 & 1.5 \\ 1.5 & 2.5 \end{bmatrix}$ 。
3. 紫色：設定樣本數為 200 ，中心點座標為 $(-3, 3)$ ，變異 - 共變異矩陣為 $\begin{bmatrix} 2 & -0.7 \\ -0.7 & 1 \end{bmatrix}$ 。

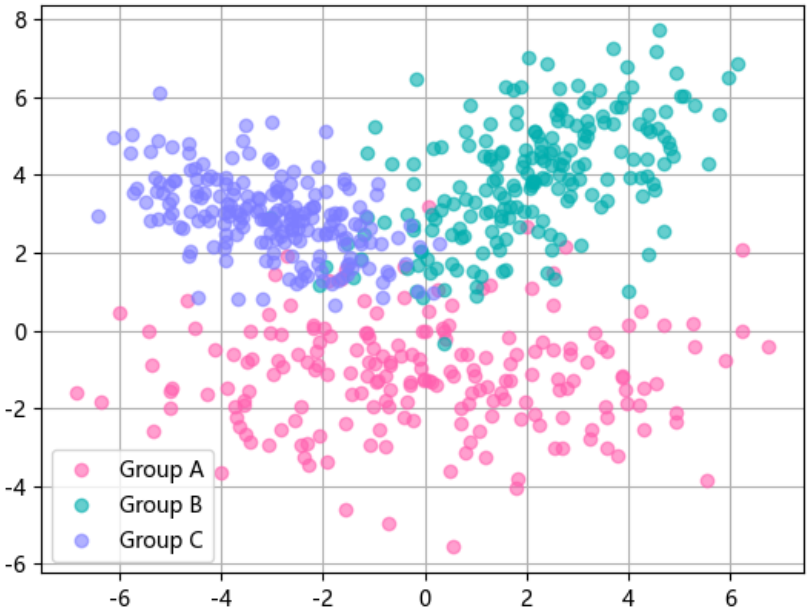


圖 12: 生出三群資料

6 邏輯斯迴歸模型處理三群資料

圖 13 為使用邏輯斯迴歸模型來處理三組資料的分群學習。

在三群資料中，邏輯斯迴歸繪製的分類邊界可以幫助理解模型如何區分三個不同的群組，這些邊界可以被用來預測新數據的分類，當使用邏輯斯迴歸繪製分類邊界時，這些線條所呈現的區域劃分能夠將特徵空間分隔成不同的區域，每個區域代表模型對應的一個類別。在三群資料中，邏輯斯迴歸繪製的分類邊界可以幫助理解模型如何區分三個不同的群組。這些邊界可以被用來預測新數據的分類。

這些分類邊界本質上是模型對特徵空間的一種投影，類似於一系列的決策邊界，它們決定了在特徵空間中不同群體的分布位置，當有新的數據點出現時，可以將其投影到這些區域中，根據其所在的位置來預測其所屬的類別。這種視覺化方式提供了對模型如何進行分類的直觀理解來更清晰地了解邏輯斯迴歸模型在處理三群資料時是如何辨別不同群組的，並能夠透過邊界的位置和形狀來推測新數據的可能分類情況。

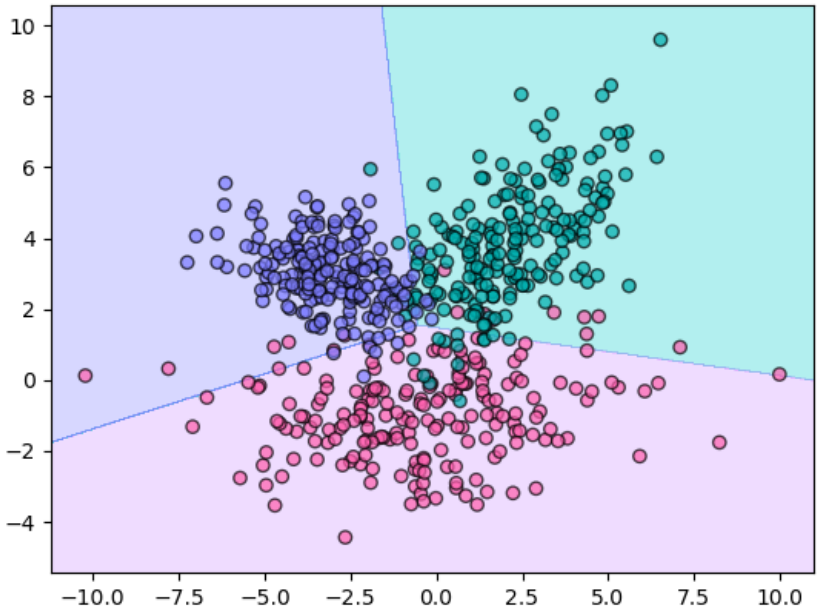


圖 13: 邏輯斯迴歸模型處理三群資料

在圖 13 中使用下列 Python 語法來用邏輯斯迴歸處理三群資料的分群學習 :

```
# 使用邏輯斯迴歸模型擬合數據
model = LogisticRegression()
model.fit(X, y)
# 繪製邊界線
h = .02
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_
   min, y_max, h))
Z = model.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
# 繪製分類區域
plt.contourf(xx, yy, Z, alpha=0.3, cmap=ListedColormap(['#CA8
   EFF', '#00CACA', '#7D7DFF']))
```

7 結論

在討論各種迴歸學習器時，可以著重於不同模型在預測和分類問題中的應用。簡單線性迴歸模型有助於捕捉特徵間的線性關係，適用於預測特定數值；增廣迴歸模型則拓展了線性迴歸，更靈活地處理非線性關係；而邏輯斯迴歸模型主要用於分類問題，能有效將數據分為不同的類別。

本文突顯了迴歸學習器在分析資料和預測未知資料中的關鍵處以及相異處，透過簡單線性迴歸模型、增廣迴歸模型和邏輯斯迴歸模型的運用，在散佈圖上呈現了資料間的分界線，並以計算結果為基礎進行更精準的資料預測，再透過計算準確率來找出哪一個模型最適合。這樣的方法不僅幫助深入理解資料間的關聯，也為預測未知資料提供了可靠的工具，增強了對資料特性和未來趨勢的洞察力。