

Model 1: 羅吉斯迴歸 (Logistic Regression) 模型

說明: 羅吉斯迴歸本質上是一個用於分類 (Classification) 的統計模型。

- **核心目標:** 它不是用來預測一個連續的數值（例如房價或溫度），而是用來預測一個事件發生的**機率**。
- **輸出結果:** 輸出值永遠介於 0 到 1 之間，代表事件發生的可能性。

在信用風險專題中，羅吉斯迴歸的任務為預測客戶在下個月發生「違約」的機率 $P(\{\text{Default}\})$ 。

➤ 羅吉斯迴歸結合了兩大核心元素：

(1) 線性組合 (Linear Combination)

模型首將所有的特徵（如年齡、信用額度、還款記錄等）乘以各自的權重/係數然後相加，得到一個線性得分 $z : z = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n$

這個 z 值可以想像成客戶的綜合風險分數，它的範圍可以是 $-\infty$ 到 $+\infty$

◦ (2) Sigmoid 轉換 (S-Curve) 由於機率必須介於 0 到 1 之間，羅吉斯迴歸會將這個線性的風險分數 z 投入 Sigmoid 函式：將 $-\infty$ 到 $+\infty$ 的值，**壓縮並轉換成** 0 到 1 之間的平滑 S 形曲線機率

(一) 輸入文件

需要使用的輸入檔案：

- **檔案名稱:** uci_default_cleaned_scaled.csv(存入名為” data” 檔案中)
- **檔案內容:** 這是已整理且標準化的版本。
- **適用性:** 該檔案已明確標註適用於 Logistic Regression、SVM、Neural Network 等需要標準化數值欄位的模型。
- **樣本數:** 最終保留 29,965 筆樣本（原始 30,000 筆，移除 35 筆重複資料）。
- **目標變數:** default payment next month (1 = 違約， 0= 正常)。

➤ (二)數據預處理與劃分 (Preprocessing & Splitting)

類別變數編碼 (One-Hot Encoding)

```
categorical_cols = ['SEX',  
'EDUCATION', 'MARRIAGE']
```

定義需要轉換為數值的類別特徵。

```
df = pd.get_dummies(df,  
columns=categorical_cols,  
drop_first=False)
```

執行 **One-Hot** 編碼。將每個類別 (如 SEX = 1) 轉換為一個二元虛擬變數 (如 SEX_1 = 1 或 0)。

特徵與目標變數定義 (X, y Definition)

```
target_col = 'default payment next  
month'
```

定義目標變數名稱。確認我們最終要預測的目標 (是否違約)。

```
X = df.drop(columns=[target_col])
```

定義特徵矩陣 X (自變數)。排除目標，只留下用於預測的 23 個特徵。

```
y = df[target_col]
```

定義目標向量 y (應變數)。客戶的實際違約標籤 0 或 1。

訓練集與測試集劃分 (Train/Test Split):

將資料劃分為訓練集樣本數: 20975 和 測試集樣本數: 8990

```
test_size=0.3,  
random_state=42
```

將 70% 數據用於訓練，保留 30% 數據用於最終評估。設置隨機數種子 42。

```
stratify=y
```

執行分層取樣。

➤ (三)模型訓練與效能評估 (Training & Evaluation)

訓練基準模型 (Training the Baseline Model)

```
log_reg_baseline = LogisticRegression(...)  
solver='liblinear'  
class_weight='balanced'  
log_reg_baseline.fit(X_train,  
y_train)
```

初始化羅吉斯迴歸模型物件。

選擇適合小型數據集的優化器。

核心參數。自動調整權重，增加少數類別(違約 1)的懲罰。

使用訓練集數據進行模型參數(係數 w 和截距 w_0)的最佳化。

效能評估 (Performance Metrics)

```
y_pred_proba =  
log_reg_baseline.predict_proba(X_test)[:,  
1]  
  
y_pred = log_reg_baseline.predict(X_test)  
  
auc_score = roc_auc_score(y_test,  
y_pred_proba)  
  
recall = recall_score(y_test, y_pred)  
  
accuracy = accuracy_score(y_test, y_pred)
```

預測機率。使用訓練好的模型對測試集 X_{test} 進行預測，並取出違約類別(1)的預測機率，這是計算 **AUC** 和信用評分的基礎。

預測類別。直接輸出模型判斷的類別標籤(0或1)，這是計算召回率和準確率的基礎。

計算 **AUC (Area Under the Curve)**。AUC 是衡量模型整體判別能力的核心指標，分數越接近1越好。

計算召回率 (**Recall Score**)。衡量在所有實際違約客戶中，模型成功預測出違約的比例。這是專案中強調的關鍵指標。

計算準確率 (**Accuracy Score**)。衡量模型預測正確的樣本佔總樣本的比例。

```

開始訓練羅吉斯迴歸模型...
模型訓練完成。

--- 基準模型效能評估（測試集） ---
AUC (Area Under the Curve): 0.7390 (判斷力指標)
召回率 (Recall Score): 0.6315 (捕捉違約客戶能力)
準確率 (Accuracy Score): 0.7283

分類報告 (Classification Report):
precision    recall   f1-score   support
0            0.88      0.76      0.81       7001
1            0.42      0.63      0.51       1989

accuracy          0.73
macro avg        0.65      0.69      0.66       8990
weighted avg     0.78      0.73      0.74       8990

```

結果分析:

AUC (Area Under the Curve)	0.7390	整體判別力指標。此數值表示模型有 73.90% 的機率能正確區分一對隨機選擇的違約客戶和正常客戶。分數在 0.7 到 0.8 之間屬於尚可接受的判別力。
召回率 (Recall Score)	0.6315	捕捉違約客戶的能力。這表示在所有實際違約的客戶中，您的模型成功預測出了約 63.15% 的客戶。這是一個相對重要的指標，因為專案要求更著重於確保高風險客戶能被抓到。
準確率 (Accuracy Score)	0.7283	整體預測正確率。模型對所有測試樣本的預測中，有 72.83% 的預測是正確的。由於您的資料存在不平衡（違約率約 22.13%），單看準確率並不夠可靠。

分類報告提供了針對「正常 (0)」和「違約 (1)」這兩個類別的詳細表現。

類別	Precision (精確率)	Recall (召回率)	f1-score (綜合指標)	support (測試集樣本數)
0 (正常)	0.88	0.76	0.81	7001
1 (違約)	0.42	0.63	0.51	1989

➤ 正常客戶 (Class 0) 的表現:

Precision 0.88 : 在模型預測為正常的客戶中，有 88% 的人確實是正常的
Recall 0.76 : 在所有實際正常的客戶中，模型成功預測出 76% 的人正常

➤ 違約客戶 (Class 1) 的表現 (重點關注)

Recall 0.6315 : 整體召回率，表示模型成功抓住 63% 的高風險客戶。

Precision 0.42 : 在模型預測為違約的客戶中，有 42% 的人最終是違約的。

風險提示：這意味著模型預測的「違約」有 $1 - 0.42 = 58\%$ 的機率是錯誤的（即誤報/偽陽性）。這是因為使用了 `class_weight='balanced'` 來提高召回率，通常會犧牲精確率，導致更多的誤報，但這在風險控管中是可以接受的權衡。

➤ (四) 模型解釋與數據輸出

係數分析 (Coefficient Analysis): 這是羅吉斯迴歸作為基準模型最重要的環節，直接解釋了模型決策的依據。

<code>coefficients = log_reg_baseline.coef_[0]</code>	提取所有特徵的係數 (\mathbf{w})。	每個係數代表該特徵對客戶對數違約勝算的貢獻。
<code>coef_df = pd.DataFrame(...), sort_values(by='Abs_Coefficient', ...)</code>	建立並按照係數的絕對值排序。	絕對值越大，影響力越強。排序後可快速找出最強的風險因子。
<code>if row['Coefficient'] > 0 else ...</code>	根據係數正負號判斷風險方向。	正係數 (如 PAYMENT_STD)：特徵值越高，增加違約風險。
		負係數 (如 EDUCATION (其他))：特徵值越高，降低違約風險。

最終數據輸出 (Final Data Output) : 最終違約機率檔案儲存至: `.data/Model1_Baseline_Final_All_Predictions.csv`

<code>all_proba = log_reg_baseline.predict_proba(X)</code>	對整個 29,965 筆樣本的特徵 X 進行預測。
<code>final_output_df = pd.DataFrame(...)</code>	整合 Client_Index、Actual_Default_Label、Predicted_Prob_Default。
<code>final_output_df.to_csv(output_csv_path, index=False)</code>	儲存最終結果 CSV 檔案。

--- 羅吉斯迴歸係數分析（特徵影響力與風險方向）---

截距 (Intercept): -0.2403

前 15 大影響特徵：

(係數的絕對值越大，影響力越大。正值 (+) 表示該特徵增加違約風險，負值 (-) 表示降低違約風險)

EDUCATION (其他)	: -0.8141 (降低違約風險 (-))
PAYMENT_STD	: 0.6265 (增加違約風險 (+))
MAX_DELAY	: 0.4340 (增加違約風險 (+))
TOTAL_PAY	: -0.3134 (降低違約風險 (-))
PAY_AMT1	: -0.2693 (降低違約風險 (-))
EDUCATION (研究所)	: 0.2587 (增加違約風險 (+))
BILL_AMT2	: 0.2536 (增加違約風險 (+))
PAY_AMT2	: -0.2506 (降低違約風險 (-))
LIMIT_BAL	: -0.1918 (降低違約風險 (-))
EDUCATION (大學)	: 0.1895 (增加違約風險 (+))
PAY_0	: 0.1888 (增加違約風險 (+))
LAST_DELAY	: 0.1888 (增加違約風險 (+))
PAY_AMT6	: -0.1818 (降低違約風險 (-))
CREDIT_UTILIZATION	: -0.1808 (降低違約風險 (-))
SEX (女性)	: -0.1793 (降低違約風險 (-))

分析：係數的絕對值越大，影響力越強。正係數增加違約風險；負係數降低違約風險。

舉幾個例子：

1	EDUCATION (其他)	-0.8141	降低違約風險 (-)	這是最強的風險緩衝因子。其他教育程度（類別 4）的客戶違約風險最低，模型給予最強負權重。
2	PAYMENT_STD [Derived]	+0.6265	增加違約風險 (+)	繳款金額的波動性越大，風險越高。這是模型認為最強的風險指標。
3	MAX_DELAY [Derived]	+0.4340	增加違約風險 (+)	歷史最嚴重延遲月數越高，風險越高。證明歷史還款行為是準確預測違約的強烈信號。
4	TOTAL_PAY [Derived]	-0.3134	降低違約風險 (-)	六期總繳款金額越高，表示還款能力強，風險越低。
5	PAY_AMT1	-0.2693	降低違約風險 (-)	最近一期繳款金額越高，風險越低。
6	EDUCATION (研究所)	+0.2587	增加違約風險 (+)	相對於其他被隱藏的基準類別，研究所教育程度的係數為正。
9	LIMIT_BAL	-0.1918	降低違約風險 (-)	信用額度越高，風險越低，符合一般金融認知。
15	SEX (女性)	-0.1793	降低違約風險 (-)	女性的違約風險略低於男性。