# HW2: Multiple Linear Regression

## 一. 作業目標：

本次作業延伸自 HW1，目標是讓同學能夠以實際資料集進行「多元線性迴歸 (Multiple Linear Regression)」的完整分析，並遵循 CRISP-DM 流程完成從資料理解、建模到評估的全過程。

## 二. 作業內容：

1. 資料來源：
   - 1.1. 至 Kaggle 選擇一個具有 10 至 20 個特徵 (features) 的公開資料集。
   - 1.2. 類型不限（資安主題）。
   - 1.3. 請明確標示資料集來源與連結。
2. 分析任務：
   - 2.1. 使用線性迴歸 (Linear Regression) 模型進行預測。
   - 2.2. 可嘗試單純線性迴歸、多元線性迴歸或 Auto Regression。
   - 2.3. 必須執行 **特徵選擇 (Feature Selection)** 與 **模型評估 (Model Evaluation)**。
   - 2.4. 結果部分需包含請提供預測圖(加上信賴區間或預測區間)
3. CRISP-DM 流程說明：
   - Business Understanding
   - Data Understanding
   - Data Preparation
   - Modeling
   - Evaluation
   - Deployment
4. AI 協助要求：
   - 4.1. 所有與 ChatGPT 的對話請以 pdfCrowd 或其他方式**須匯出為 PDF**。
   - 4.2. 請**使用 NotebookLM** 對網路上同主題的解法進行研究，並撰寫一份 100 字以上的摘要，放入報告中。
   - 4.3. 請在報告中明確標示「GPT 輔助內容」與「NotebookLM 摘要」。
5. 繳交內容：
   - 5.1. 主程式：5114050013_hw2.py /.ipynb
   - 5.2. 報告檔： PDF，需包含以下內容：
     - ① 按照 CRISP-DM 說明的分析流程
     - ② GPT 對話過程（pdfCrowd 匯出）
     - ③ NotebookLM 研究摘要
     - ④ 網路上主流或更優解法之比較與說明

   註 1：以上檔案與資料夾，**請壓縮為學號命名的一個zip**（例如：5114050013_hw2.zip）上傳。

   註 2：(optional) 若上傳至 GitHub，或是以colab撰寫，需提供連結，並在 README.md 中整理流程與成果。

三. 評分標準：

1. 文件說明（50%）
   1.1. CRISP-DM 流程完整且邏輯清楚（25%）
   1.2. 包含 GPT 對話與 NotebookLM 摘要（15%）
   1.3. 有明確說明資料集來源與研究脈絡（10%）
2. 結果呈現（50%）
   2.1. 模型正確可執行，具特徵選擇與評估（25%）
   2.2. 結果合理、美觀且具有說服力（15%）
   2.3. 呈現出Kaggle名次(若有) / 預測結果評估(預測圖、評估指標)（10%）

**[Datasets Source URL]**
https://www.kaggle.com/datasets/atharvasoundankar/global-cybersecurity-threats-2015-2024

**[my GitHub website URL]**
https://github.com/candice-wu/Cybersecurity_HW_02_Multiple_Linear_Regression.git

**[my Streamlit Demo-site URL]**
https://hw02-multiple-linear-regression.streamlit.app/

## 💻 網路安全威脅財務損失預測 (Cybersecurity Threat Financial Loss Prediction)

Goal：預測各項資安事件造成的財務損失（Financial Loss in Million $）
Framework：CRISP-DM + Streamlit Deployment

### 一. 資料集來源與連結

◉ Dataset Source：Kaggle

◉ Subject：
Global Cybersecurity Threats (2015 ~ 2024)

◉ Website link：
https://www.kaggle.com/datasets/atharvasoundankar/global-cybersecurity-threats-2015-2024

◉ Datasets Introduction：

① Scope：
The Global Cybersecurity Threats Dataset (2015 ~ 2024) provides extensive data on cyberattacks, malware types, targeted industries, and affected countries.
It is designed for threat intelligence analysis, cybersecurity trend forecasting, and machine learning model development to enhance global digital security.

② Data Overview - Features｜Target (Please kindly refer to below table)

✳ Target：設定「Financial Loss (in Million $)」作為目標變數 (y)，其他欄位來預測經濟損失，目的：探討「哪些因素會影響資安事件的損失金額」。

✳ Features：除了 Target 的其它 9 個欄位。

| Column Name | Data Type | Description | Sample Value |
|---|---|---|---|
| Country | object | 發生國家 | China, UK, Germany |
| Year | int | 發生年份 (2015–2024) | 2019 |
| Attack Type | object | 攻擊類型 | Ransomware, DDoS, Phishing |
| Target Industry | object | 攻擊目標產業 | IT, Finance, Retail |
| Financial Loss (in Million $) | float | 經濟損失 (百萬美金) | 80.53 |
| Number of Affected Users | int | 受影響用戶數 | 773169 |
| Attack Source | object | 攻擊來源 | Hacker Group, Nation-state |
| Security Vulnerability Type | object | 弱點類型 | Zero-day, Weak Passwords |
| Defense Mechanism Used | object | 防禦機制 | VPN, Firewall, Antivirus |
| Incident Resolution Time (in Hours) | int | 事件處理時間 (小時) | 63 |

## 二. CRISP-DM 實作架構

| 流程階段 | 目標 | 主要任務 | 對應方法 / Python 模組 |
|---|---|---|---|
| 1. 業務理解<br>(Business Understanding) | 明確定義分析目標與應用場景 | 1. 定義研究問題：「哪些因素最影響資安事件的經濟損失？」<br>2. 說明資料應用價值與決策意義 | N/A |
| 2. 數據理解<br>(Data Understanding) | 了解資料結構與特徵分布 | 1. 使用 df.info()、df.describe()、df.isnull().sum()<br>2. 使用 seaborn 與 matplotlib 畫出關鍵變數分布、箱型圖、熱力圖 | pandas<br>seaborn<br>matplotlib |
| 3. 數據準備<br>(Data Preparation) | 清理、轉換與特徵工程 | 1. 缺失值處理、Outlier 檢查<br>2. 類別特徵轉為 One-Hot Encoding<br>3. 數值標準化 (StandardScaler)<br>4. 訓練/測試資料切分 (train_test_split) | pandas<br>sklearn.preprocessing |
| 4. 建模<br>(Modeling) | 建立預測模型 | 1. 使用 LinearRegression 建立多元線性模型<br>2. 可嘗試 Auto Regression (若以年份為時間序列)<br>3. 使用 RFE 進行特徵選擇 | sklearn.linear_model<br>sklearn.feature_selection |
| 5. 評估<br>(Evaluation) | 檢驗模型表現與合理性 | 1. 計算 $R^2$、RMSE、MAE<br>2. 繪製「實際 vs 預測」散點圖、殘差圖<br>3. 繪製含信賴區間的預測圖 | sklearn.metrics<br>matplotlib |
| 6. 部署<br>(Deployment) | 展示可執行、具互動性成果 | 1. 將訓練好的模型儲存 (joblib.dump)<br>2. 使用 Streamlit 實作互動式預測應用：<br>① 使用者輸入攻擊屬性 → 即時預測財務損失<br>② 顯示信賴區間與圖形化 | joblib<br>streamlit |

這是一個基於機器學習的專案，旨在預測網路安全事件可能造成的財務損失。
專案採用 CRISP-DM（Cross-Industry Standard Process for Data Mining）流程方法論，從商業理解到模型部署，提供了一個完整的資料科學專案範例。

使用者可以透過一個互動式的 Streamlit 網頁應用程式，輸入假設的攻擊情境，來預測潛在的財務損失，並深入探索資料與模型。

1. 商業理解（Business Understanding）
   - ◉ 目標說明
     - ✳ 探討哪些因素會影響資安事件造成的財務損失。
       即隨著全球數位化轉型，網路安全事件頻傳，對企業造成的財務衝擊也日益嚴重。本專案的主要商業目標是建立一個數據驅動的預測模型，以協助企業或組織評估不同網路安全威脅事件可能帶來的財務損失（以百萬美元計）。

     - ✳ 建立可預測 Financial Loss 的迴歸模型。
       ① 更精準地評估資安風險。
       ② 優先處理和分配資源給可能造成重大損失的威脅類型。
       ③ 為資安保險、預算規劃和投資決策提供量化依據。

   - ◉ 任務重點
     - ✳ 明確定義 Target：Financial Loss (in Million $)。
     - ✳ 其餘欄位為 features（9 欄），需進行類別與數值型特徵處理。

2. 資料理解（Data Understanding）
   - ◉ 任務
     - ✳ 載入資料集（Global_Cybersecurity_Threats_2015-2024.csv）
       從 2015 年到 2024 年間的全球網路安全威脅事件記錄。
     - ✳ 資料特徵（Features）： 資料集包含多種數值和類別特徵，如：
       ① Attack Type: 攻擊類型 (e.g., DDoS, Malware, Phishing)
       ② Country: 攻擊發生的國家
       ③ Sector: 受攻擊的產業別
       ④ Number of Affected Users: 受影響的使用者數量
       ⑤ Incident Resolution Time (in Hours): 事件解決所需時間（小時）
     - ✳ 目標變數（Target）：Financial Loss (in Million $): 財務損失（百萬美元）
     - ✳ 瞭解欄位資訊、缺失值與統計摘要
     - ✳ 視覺化資料分布與關聯性
       在 Streamlit.app 的「分析頁面」中，「資料概覽」和「特徵分佈分析」區塊提供了對資料的深入探索，包括資料偏度分析、各特徵的分佈直方圖、盒鬚圖等。

◉ 程式碼

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# 載入資料集
df = pd.read_csv("Global_Cybersecurity_Threats_2015-2024.csv")
print("資料集前五筆：")
display(df.head())
```

Python

```python
print("資料集資訊：")
df.info()
```

Python

```python
print("數值特徵統計摘要：")
display(df.describe())
```

Python

3. 資料預處理（Data Preprocessing）

◉ 任務

此階段由 prepare_data.py 腳本負責，主要執行以下步驟：

| | |
|---|---|
| 1. 載入資料 | 從 CSV 檔案載入資料集，包括缺失值處理。 |
| 2. 特徵工程 | 獨熱編碼（One-Hot Encoding 類別特徵）將所有類別特徵（如 Attack Type, Country）轉換為數值格式，以便機器學習模型能夠處理。 |
| 3. 資料標準化 | 使用 StandardScaler 對所有數值特徵進行標準化，使其具有零均值和單位變異數。這一步驟對於線性模型和 RFE 的穩定性至關重要。 |
| 4. 資料分割（Train/Test） | 將處理後的資料集以 80/20 的比例分割為訓練集和測試集。此過程中使用固定的 random_state 以確保結果的可重現性。 |
| 5. 儲存交付物 | |

◉ 程式碼

```python
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
import joblib

df = pd.read_csv('Global_Cybersecurity_Threats_2015-2024.csv')

# 設定 target 與 features
target = "Financial Loss (in Million $)"
X = df.drop(columns=[target])
y = df[target]

# 類別欄位轉換
X = pd.get_dummies(X, drop_first=True)

# 數值標準化
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# 訓練/測試集切分
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)

# Save the data and scaler
np.save('X_train.npy', X_train)
np.save('X_test.npy', X_test)
np.save('y_train.npy', y_train)
np.save('y_test.npy', y_test)
joblib.dump(scaler, 'scaler.pkl')
joblib.dump(X.columns, 'feature_names.pkl')

# Save full processed X and y for statsmodels
np.save('X_full_processed.npy', X)
np.save('y_full_processed.npy', y)
print('✅ 資料準備完成，相關檔案已儲存。')
```

4. 模型建立（Modeling）

◉ 任務

此階段由 train_model.py 腳本負責。建立兩個互補的迴歸模型：

① Scikit-learn 線性迴歸 + RFE

- 遞歸特徵消除 (RFE)：首先，我們使用 RFE 來自動篩選出對預測財務損失最重要的 10 個特徵。
- 線性迴歸 (Linear Regression)：接著，我們使用一個標準的線性迴歸模型，僅在 RFE 篩選出的特徵上進行訓練。這個模型 (cyber_risk_model.pkl) 主要用於產生最終的預測值。

② Statsmodels OLS 模型

- 使用 statsmodels 函式庫建立了一個普通最小二乘法 (OLS) 模型。此模型 (statsmodels_model.pkl) 的優勢在於提供詳細的統計摘要，包括特徵的 p-value、信賴區間等。
- 在本專案中，它主要用於計算預測值的 95% 預測區間，並在「特徵重要性」分析中提供係數參考，額外提供可以隨機調整預測區間的功能。

◉ 程式碼

```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.feature_selection import RFE
import joblib
import statsmodels.api as sm
import pandas as pd

# Load the data for sklearn model
X_train = np.load('X_train.npy', allow_pickle=True)
y_train = np.load('y_train.npy', allow_pickle=True)

# Create a Linear Regression model
model = LinearRegression()

# Use Recursive Feature Elimination (RFE) to select the best features
rfe = RFE(model, n_features_to_select=10)
X_train_rfe = rfe.fit_transform(X_train, y_train)

# Train the sklearn model on the selected features
model.fit(X_train_rfe, y_train)

# Save the trained sklearn model and the RFE selector
joblib.dump(model, 'cyber_risk_model.pkl')
joblib.dump(rfe, 'rfe.joblib')

print("Sklearn model trained and saved successfully.")
```

```
# --- Fit and Save Statsmodels OLS Model ---

# Load full processed X and y for statsmodels
X_full_processed = np.load('X_full_processed.npy', allow_pickle=True)
y_full_processed = np.load('y_full_processed.npy', allow_pickle=True)
feature_names = joblib.load('feature_names.pkl')

# Create DataFrame for statsmodels and ensure numeric types
X_sm = pd.DataFrame(X_full_processed, columns=feature_names).astype(float)
y_sm = pd.Series(y_full_processed).astype(float)

# Add a constant to the X for statsmodels (for intercept)
X_sm = sm.add_constant(X_sm)

# Get the names of the features selected by RFE
selected_feature_names_rfe = feature_names[rfe.support_].tolist()
if 'const' not in selected_feature_names_rfe:
    selected_feature_names_rfe.insert(0, 'const')

# Filter X_sm to include only the RFE-selected features
X_sm_selected = X_sm[selected_feature_names_rfe]

# Fit statsmodels OLS model
sm_model = sm.OLS(y_sm, X_sm_selected).fit()

# Save the statsmodels model
joblib.dump(sm_model, 'statsmodels_model.pkl')

print("Statsmodels OLS model trained and saved successfully.")
```

5. 模型評估（Evaluation）- 模型評估與結果視覺化

◉ 任務

模型的評估在 Streamlit 應用程式的「分析頁面」中進行，主要包含以下幾個部分：

✳ 迴歸指標 - 評估模型表現

① R-squared ($R^2$)：解釋模型對目標變數變異性的解釋程度

② Root Mean Squared Error (RMSE)：衡量預測值與實際值之間的平均誤差幅度

③ Mean Absolute Error (MAE)：另一種誤差的衡量方式，較不受異常值影響

✳ 圖表建議

① 實際 vs. 預測圖：一個散點圖，用於比較實際損失與模型預測損失的一致性

② 殘差圖：用於檢查誤差是否隨機分佈，是評估模型假設的重要工具

③ 混淆矩陣：雖然這是迴歸問題，但將連續的損失值分為「高、中、低」三個等級，並建立一個互動式的混淆矩陣。讓使用者可以從「分類」的角度評估模型在不同損失等級上的預測準確度，並可依特定特徵進行篩選分析

◉ 程式碼

```
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
import numpy as np

# Load test data
X_test = np.load('X_test.npy', allow_pickle=True)
y_test = np.load('y_test.npy', allow_pickle=True)

# Load model and rfe
model = joblib.load('cyber_risk_model.pkl')
rfe = joblib.load('rfe.joblib')

# Transform test data
selected_X_test = rfe.transform(X_test)

# Generate predictions
y_pred = model.predict(selected_X_test)

r2 = r2_score(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
mae = mean_absolute_error(y_test, y_pred)

print("### 模型評估指標 ###")
print(f"- **R-squared (R²):** {r2:.3f}")
print(f"- **Root Mean Squared Error (RMSE):** {rmse:.3f}")
print(f"- **Mean Absolute Error (MAE):** {mae:.3f}")

# Scatter plot of actual vs. predicted
plt.figure(figsize=(8, 6))
sns.scatterplot(x=y_test, y=y_pred)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--', lw=2)
plt.xlabel("Actual Financial Loss (Million $)")
plt.ylabel("Predicted Financial Loss (Million $)")
plt.title("Actual vs. Predicted Financial Loss")
plt.show()
```

6. 部署（Deployment）

本專案的最終產出是部署在 Streamlit 上的互動式網頁應用程式 (5114050013_hw2.py)

◉ 任務

✳ 將訓練好的模型儲存（joblib.dump）

✳ 使用「Streamlit」建立互動式預測應用。

✳ 應用程式功能：

▸ 預測頁面

在側邊欄輸入各種攻擊事件的參數（如年份、攻擊類型、影響用戶數等），點擊按鈕後，應用程式會立即回傳預測的財務損失金額，並以圖表形式展示其 95% 預測區間。

▸ 分析頁面（提供一個功能豐富的儀表板）

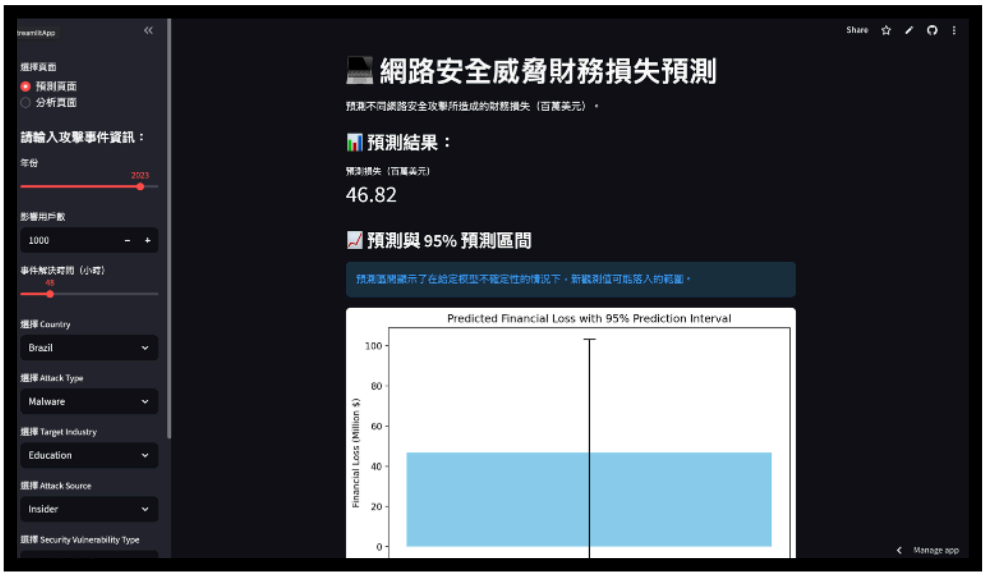
① 概覽資料集的統計特性與分佈。

② 探索不同特徵之間的關係以及它們對財務損失的影響。

③ 查看模型的詳細性能指標與評估圖表。

④ 分析 RFE 所選出的重要特徵。

⑤ 透過互動式混淆矩陣，深入了解模型在特定情境下的分類表現。




## 三. Streamlit 互動頁面設計範例（邏輯結構）

3-1. 以下是建議的頁面結構與互動流程，可直接照此在 .py 或 .ipynb 中實作。

| 塊 | 功能說明 | Streamlit 元件建議 |
|---|---|---|
| Header 區 | 顯示標題與專案說明 | st.title()<br>st.markdown() |
| Sidebar 區 | 使用者輸入控制 | st.sidebar.selectbox()<br>st.sidebar.slider()<br>st.sidebar.number_input() |
| Main 區（預測結果） | 顯示模型預測與評估指標 | st.metric()<br>st.write()<br>st.pyplot() |
| 圖表區 | 視覺化預測結果與信賴區間 | matplotlib / plotly |
| 模型資訊區 | 顯示 $R^2$、RMSE、MAE 等指標 | st.table() 或 st.write() |

3-2. Wireframe

---

💻 Cybersecurity Threat Loss Prediction App

---

[ Sidebar ]
Year: [2023]
Attack Type: [Ransomware]
Target Industry: [Finance]
...

---

預測損失金額： $85.37 Million
$R^2$ = 0.82, RMSE = 6.13
[ 散點圖 / 信賴區間圖 / 殘差圖 ]

---

# Appendix

## A. NotebookLM 研究摘要（至少 100 字）

- 搜尋主題：「Multiple Linear Regression for Cybersecurity Threat Financial Loss Prediction」
- 重點：網路上主流或更優解法之比較與說明
- 摘要：

在網路安全財務損失預測的領域中，多元線性迴歸 (Multiple Linear Regression, MLR) 通常作為基線模型使用。然而，研究結果顯示，線性迴歸在預測財務損失方面的能力非常有限（例如，$R^2$ 值僅約 –.04），難以捕捉資安事件與損失之間的複雜非線性關係。

主流或更優的解決方案主要集中在進階機器學習（ML）和複雜的統計方法：
1. 集成式機器學習模型：XGBoost 被證明是預測財務損失的最佳模型，其預測準確度（$R^2 = 0.74$）顯著高於線性迴歸和 Random Forest ($R^2 = 0.66$)，顯示出它在處理高維度、混合型資料和複雜互動關係方面的卓越能力。

2. 進階迴歸技術：為了獲得更精確的模型並進行特徵選擇，研究者採用了懲罰性迴歸方法，如 Lasso、Ridge 和 Elastic Net。例如，Elastic Net 在預測特定損失類型（如主要應對成本 PRC）時，表現優於標準 OLS 模型。此外，分位數迴歸也被提出用於捕捉網路損失成本分佈中的異質性，這是傳統線性迴歸無法實現的。

3. 複雜統計與精算模型：由於網路風險具有重尾分佈和高度相依性的特徵，超越簡單迴歸和標準分佈是必要的。因此，極值理論 (EVT) 應用於建模極端損失事件，Copula 模型則用於精確描述不同損失成分或攻擊率之間的非線性依賴結構，而流行病學模型則用於模擬風險在網路中的傳播與演化。

總結來說，雖然 MLR 可作為起點，但為了提供準確、實時的風險管理洞察和預測，XGBoost 等複雜的非線性 ML 模型，以及整合了 EVT 和 Copula 理論的精算框架，是目前公認的主流或更優解法

## B. ChatGPT 對話記錄摘要（PDFCrowd 其它方式匯出 PDF）

請參閱「人工智慧與資訊安全_HW02_與 chatGPT 對話記錄.pdf」。