

Simple Linear Regression using CRISP-DM

基於 CRISP-DM 的簡單線性回歸專案

1. Project Overview 專案概要

This project demonstrates a Simple Linear Regression process following the CRISP-DM framework. Users can adjust model parameters (a, b, noise, data points) and deploy the model via Streamlit or Flask.

本專案示範以 CRISP-DM 流程進行簡單線性回歸，使用者可調整模型參數 (a, b, 噪音、資料量)，並透過 Streamlit 或 Flask 部署。

2. CRISP-DM Steps 專案流程

Business Understanding 業務理解	Define problem and project goal 定義問題與目標
Data Understanding 資料理解	Explore generated synthetic data 探索產生之模擬資料
Data Preparation 資料準備	Split data into training/testing sets 分割訓練與測試資料
Modeling 建模	Apply Linear Regression ($y = ax + b$) 建立線性回歸模型
Evaluation 評估	Use MSE and R^2 metrics 使用 MSE 與 R^2 評估模型
Deployment 部署	Deploy via Streamlit / Flask 透過 Streamlit 或 Flask 部署

3. Implementation 實作重點

- Adjustable parameters: slope (a), intercept (b), noise, number of points
- Frameworks: Streamlit & Flask
- PDF export feature
- Visualization: regression line, scatter plot, residuals

實作重點包括：可調整參數、使用 Streamlit 與 Flask、支援 PDF 匯出、與視覺化圖表。

4. Deployment Guide 部署說明

- Streamlit: Run with ``streamlit run app.py``
- Flask: Run with ``flask run`` or deploy on Render/Railway
- GitHub Actions: CI/CD setup for both frameworks

部署可使用 Streamlit、Flask 或 GitHub Actions 自動化流程。

5. Project Structure 專案結構

```
`` linear_regression_crispdm/      app.py # Streamlit App      app_flask.py # Flask
App      core.*.md # CRISP-DM documentation      requirements.txt
.github/workflows/ # GitHub Actions      Procfile # Flask deployment      .gitignore ``
```

6. How to Use 使用方式

- 1 Open terminal in VS Code
- 2 Modify slope, intercept, noise, and data points
- 3 Run app and visualize model
- 4 Export results to PDF

開啟 VS Code 調整參數後執行應用程式，即可視覺化模型並匯出 PDF 報告。

7. Credits & Versions 版本與致謝

Developed with using Python, Streamlit, Flask, and fpdf.

版本：1.0.0 | 作者：ChatGPT (OpenAI)