

Gemini 輔助內容

前言

本專案從資料集的選擇、程式碼的開發、分析報告的撰寫，到最終的 Streamlit 應用程式部署，全程在 Google Gemini 的輔助下完成。本章節旨在記錄並總結在此次人機協作過程中，我是如何利用 Gemini 作為一個強大的開發與分析夥伴，以及這個協作模式如何極大地提升了專案的執行效率與深度。

1. 互動式開發過程摘要

本次作業的完成過程，可以被視為一個敏捷且高效的「對話式開發」(Conversational Development) 流程。整個專案的推進並非一次性的指令，而是透過與 Gemini 的一系列漸進式對話來逐步建構，完整體現了從無到有的過程。

根據 hw2-log.txt 的記錄，其開發流程如下：

1. 專案啟動與規劃 (Prompt 1-2): 我首先向 Gemini 提出高層次需求，要求推薦適合進行多元線性回歸分析的資料集。Gemini 提供了三個選項並附上詳細理由，我據此選定了「紅酒品質」資料集。接著，我要求 Gemini 依據 CRISP-DM 框架，撰寫專案的「商業理解」與「資料理解」部分，為整個專案確立了清晰的分析方向。
2. 核心應用程式搭建 (Prompt 3): 在確立目標後，我指令 Gemini 開始實作，要求建立 `app.py` 檔案並撰寫「資料準備」階段的 Python 程式碼。Gemini 產出了包含離群值處理、特徵標準化和資料切分的完整、可執行的 Streamlit 程式碼。
3. 模型功能迭代增強 (Prompt 4-7): 在應用程式的基礎上，我透過一系列指令，逐步為其增加新功能，每一項都是一個獨立的分析模組：
 - 模型訓練: 加入基礎的多元線性回歸模型。
 - 特徵選擇: 引入 SelectKBest 讓模型更具解釋性。
 - 模型評估: 新增 R^2 、MAE、MSE 等關鍵效能指標的計算與分析。
 - 視覺化分析: 產生預測值與殘差的散佈圖，使模型評估更為直觀。
4. 文件與研究報告生成 (Prompt 8-9): 最後，我要求 Gemini 根據已完成的所有步驟，撰寫完整的 CRISP-DM.txt 報告，並額外生成一份關於多元線性回歸優化方法的 NotebookLM.txt 研究摘要。

這個過程顯示，Gemini 不僅僅是一個程式碼生成器，更是一個能夠參與專案規劃、執行、評估到文件產出的全程夥伴。

1. 我的 Gemini 應用策略：從指令到成果

在這次協作中，我扮演的角色更像是一位「專案經理」或「產品負責人」，而 Gemini 則是高效的「AI 開發者與資料分析師」。我的主要策略如下：

- 拆分任務，循序漸進: 我沒有一開始就提出「幫我完成所有作業」的模糊指令，而是遵循 CRISP-DM 的清晰框架，將複雜的任務拆解成一個個具體、可執行的步驟。hw2-log.txt 中的每一條 prompt 都對應了分析流程中的一個邏輯單元。
- 提供上下文，明確目標: 在每一次請求中，我都讓 Gemini 基於已存在的 `app.py` 檔案進行修改與擴充。這種提供上下文的方式，讓 Gemini 能夠準確理解我的意圖，並在現有基礎上進行無縫的功能疊加。
- 專注於「What」，而非「How」: 我將精力集中在定義「需要完成什麼」(What)，例如「幫我做特徵選擇」、「幫我計算評估指標」。而具體的「如何實現」(How)，包括選擇哪個函式庫、撰寫 Streamlit 的版面配置、處理程式錯誤（如日誌中記錄的一次 `replace` 指令失敗後，Gemini 自動切換策略），都由 Gemini 高效完成。

這種人機協作模式，讓我能從繁瑣的語法查詢和程式碼撰寫中解放出來，更專注於專案的宏觀架構、分析邏輯與結果解讀，從而極大地加速了從概念到成品的轉換過程。

1. 與 NotebookLM 摘要的整合建議

Gemini 產出的 NotebookLM.txt 研究摘要，不僅是一份獨立的文件，它更為本專案的下一步優化提供了清晰的藍圖。摘要中提到的「主流進階方法」與「模型優化建議」可以直接轉化為 `app.py` 的新功能。

整合建議如下：

1. 實現進階迴歸模型:

- 在 Streamlit 應用中新增一個模型選擇器（如下拉式選單），讓使用者可以在「線性迴歸」、「嶺迴歸 (Ridge)」和「套索迴歸 (Lasso)」之間切換。
- 執行後，並排比較不同模型的 R^2 和 RMSE，讓使用者直觀感受正規化技術對模型效能的影響。

2. 引入特徵工程選項:

- 在「特徵選擇」區塊後，新增一個「特徵工程」區塊。

- 提供核取方塊 (Checkbox)，允許使用者動態生成「多項式特徵」（例如，對 alcohol 產生二次方項）或「交互作用項」（例如，alcohol * sulphates）。
- 重新訓練模型並更新評估指標，驗證這些新特徵是否能如摘要所說，有效捕捉非線性關係並提升模型表現。

3. 將問題重新定義為分類任務:

- 在應用程式的最後，新增一個「嘗試分類模型」的實驗性區塊。
- 將 quality 分數轉換為類別（例如，quality ≥ 7 為「優質」，quality ≤ 4 為「劣質」，其餘為「普通」）。
- 訓練一個簡單的分類模型（如邏輯迴歸），並展示其準確率 (Accuracy) 和混淆矩陣 (Confusion Matrix)，驗證 NotebookLM 摘要中「重新定義問題」的策略是否可行。

透過以上整合，本專案將從一個單一模型的分析，演變為一個功能豐富、可供比較和探索的互動式機器學習實驗平台，完美展現了從理論研究到實踐應用的閉環。