HW3:

Source Reference

This project builds upon patterns and datasets related to the Spam Email problem from Chapter 3 of the Packt repository below. We used it to expand the preprocessing steps and add richer visualization work (step outputs, metrics, and CLI/Streamlit views).

https://github.com/PacktPublishing/Hands-On-Artificial-Intelligence-for-Cybersecurity.git

Using openspec and AI coding CLI to finish this project

Requirements：

1. need a github
   https://github.com/huanchen1107/2025ML-spamEmail

2. need a Demo site
   https://2025spamemail.streamlit.app/

[my GitHub websit]
https://github.com/candice-wu/Security_HW_03_Classification_Spam-Email

[my Streamlit Demosite]
https://hw03-classification-spam-email.streamlit.app/

[執行模式，採二階段開發]

Phase 1：請 Gemini 根據 openspec 專案架構幫我產生初版

Phase 2：請 Gemini 升級我的專案

註 1：1st 是 openspec 搭配 GitHub Copilot，但 GitHub Copilot 不太聰明，執行過程不太像老師上課所教，及 AI 超元域的網路教學，放棄使用。

註 2：openspec 內建預設的 AI coding CLI 雖沒有 Gemini CLI，但我預選 GitHub Copilot 後，嘗試在 Termina 呼叫 Gemini，意外地發現 Gemini 也能依照 openspec 架構生成相關文件（如同老師上課所教，及 AI 超元域的網路教學），介紹如下：

[Phase 1：請 Gemini 根據 openspec 專案架構幫我產生初版]

1. 啟動 openspec 與 Gemini CLI 並請求產生專案提案的企劃案文件

圖 1: openspec init

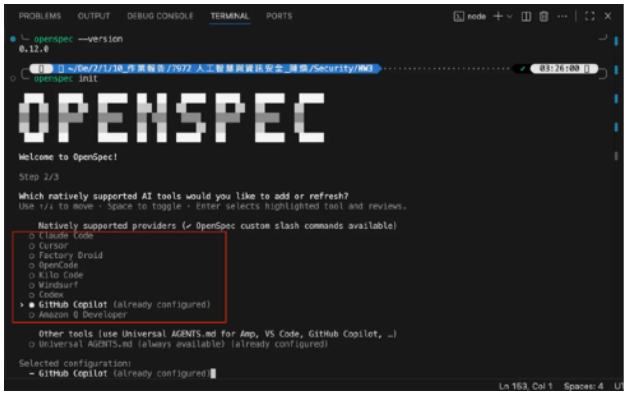


圖 2: Select GitHub Copilot

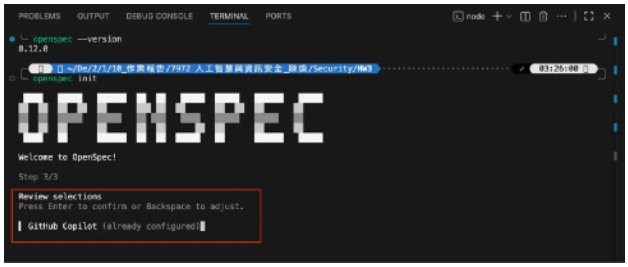


圖 3: Double confirm

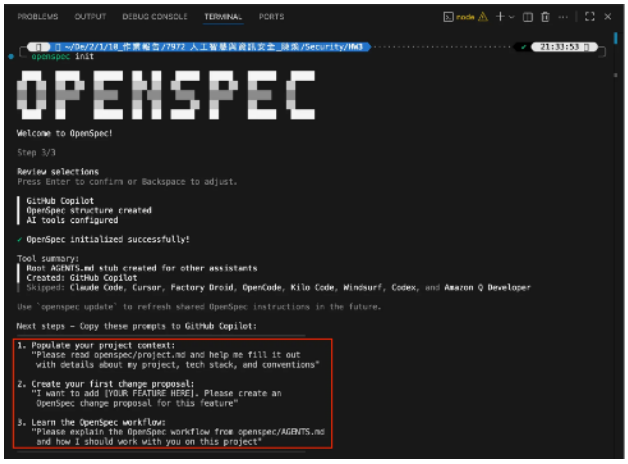


圖 4: Openspec 要求轉貼三個指令到 AI coding CLI

圖 5: 啟動 (呼叫) Gemini CLI

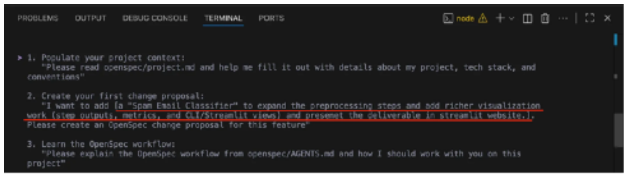


圖 6: 將 openspec 第二個指令的中括號 [YOUR FEATURE HERE] 改成我的專案內容 [a "Spam Email Classifier" to expand the preprocessing steps and add richer visualization work (step outputs, metrics, and CLI/Streamlit views) and presenet the deliverable in streamlit website.]。

如此，Gemini 更清楚我的專案需求，並依 openspec 架構自動產生專案提案文件：proposal.md、design.md、tasks.md、specs/spam-classifier

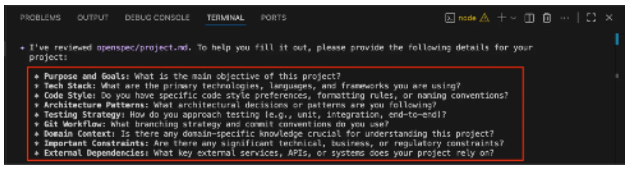


圖 7: Gemini CLI 列出 9 項問題要我澄清並回覆，以便它能更清楚掌握我的專案要求

圖 8: 我回覆給 Gemini CLI 的內容 (專案規範要求)

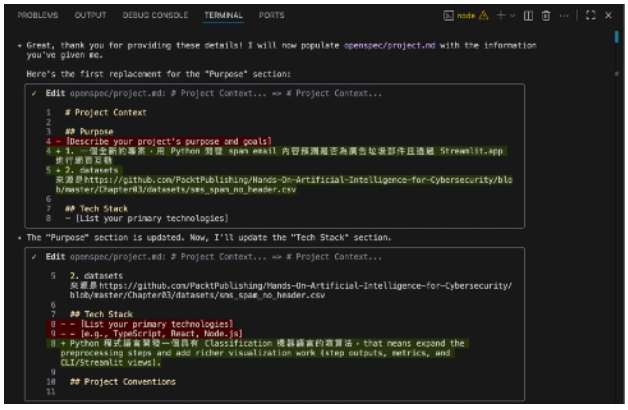


圖 9: Gemini CLI 接著會開始讀取分析並自動生成各資料夾或文件以符合專案需求，過程中會逐一再次詢問你所回覆的 9 項答覆 (專案規範要求) 的項次1、2

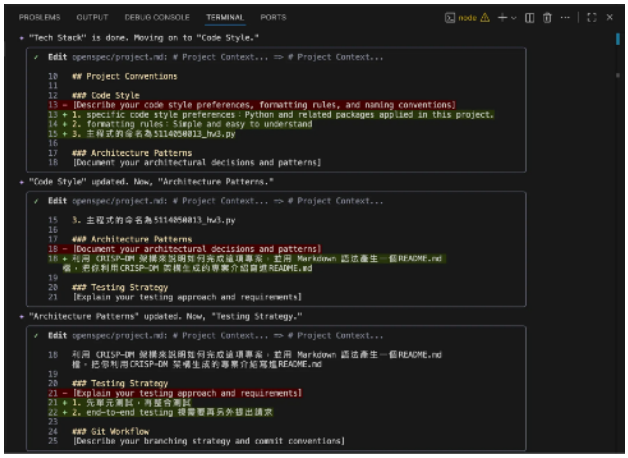
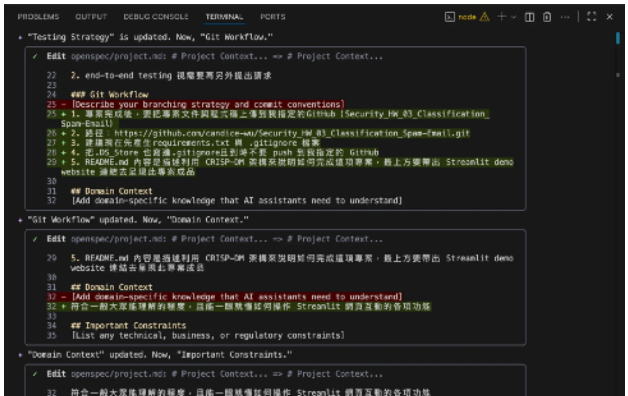


圖 10: Gemini CLI 逐一要我確認我回覆的 9 項答覆 (專案規範要求) 的項次 3～5



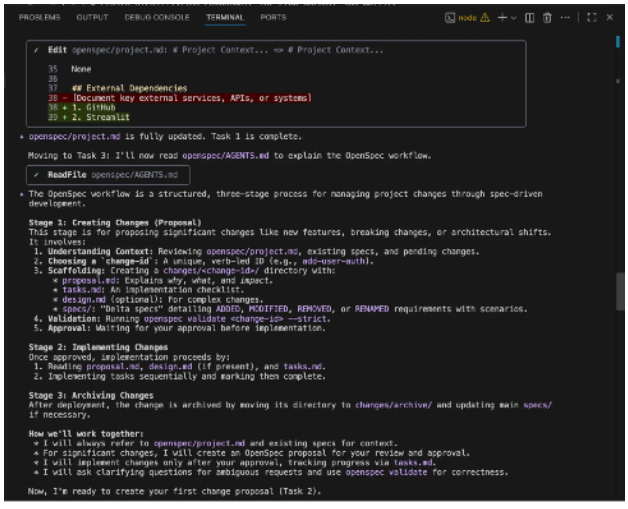
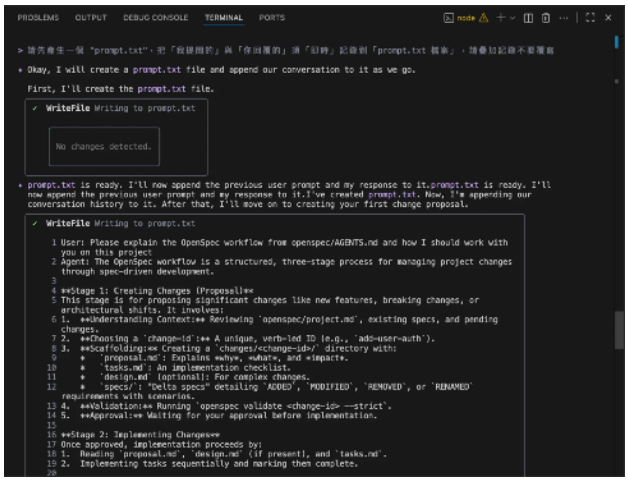


圖 11: Gemini CLI 逐一要我確認我回覆的 9 項答覆 (專案規範要求) 的項次 6～8

圖 12: Gemini CLI 逐一要我確認我回覆的 9 項答覆 (專案規範要求) 的項次 9。接著，它表示 Task 1 已完成，並主動先跳到 Task 3 解釋 OpenSpec workflow，最後表示說要開始進行 Task 2



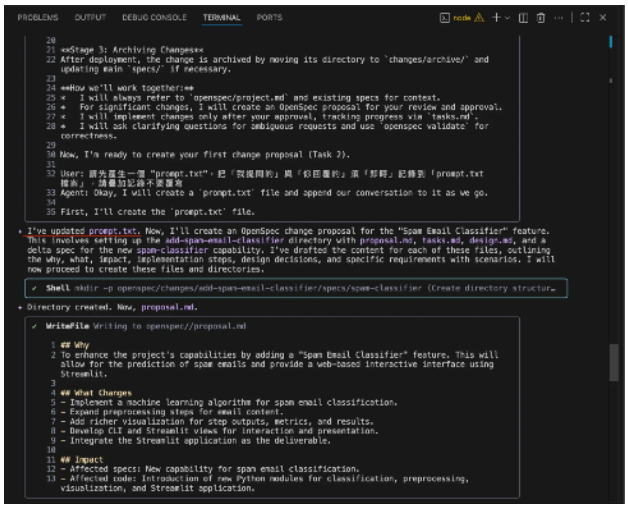
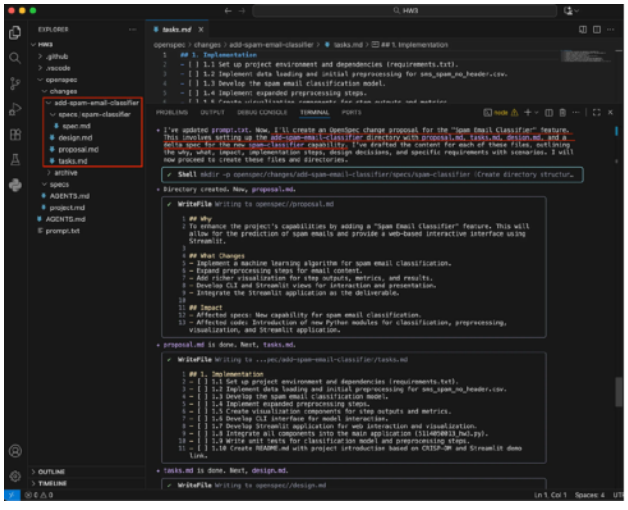


圖 13: Gemini CLI 接下來要進行 Task 2（create your first change proposal），但我先請它幫我先做對話紀錄並寫進 prompts.txt

圖 14: Gemini CLI 幫忙生成對話記錄檔，也很聰明地把前面的對話自動回補到 prompts.txt。



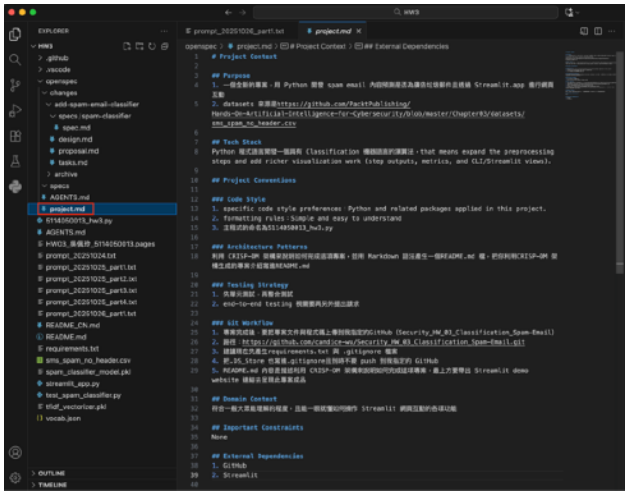


圖 15: Gemini CLI 完成對話記錄檔，自動回到 Task 2 依 openspec 架構去自動產生 OpenSpec change proposal，自動產生 add-spam-email-classifier 資料夾，底下有 proposal.md, tasks.md, design.md, and a delta spec，即自動地生成專案提案

圖 16: 之前我回覆的9項專案規範要求，Gemini CLI 則產生一份 project.md 去記錄

註：可透過以下指令請 Gemini CLI 確認專案內容

指令	用途	常見參數 / 備註
openspec list	列出目前所有「變更 (changes)」資料夾和內容狀態（例如：哪些 change 尚未審核、哪些已完成）	
openspec show <資料夾名稱>	定義一個新的變更（例如 “add-spam-classifier”）後用來顯示詳細內容：Proposal、Tasks、Spec design 等	
openspec validate <資料夾名稱>	驗證指定變更的格式與結構是否正確（例如是否缺少 proposal.md、tasks.md、specs 目錄等）	
`openspec archive [--yes	-y]`	封存（archive）已完成的變更並更新主規格（Specs）為最新狀態。

2. Gemini CLI 開始執行自動化程式開發

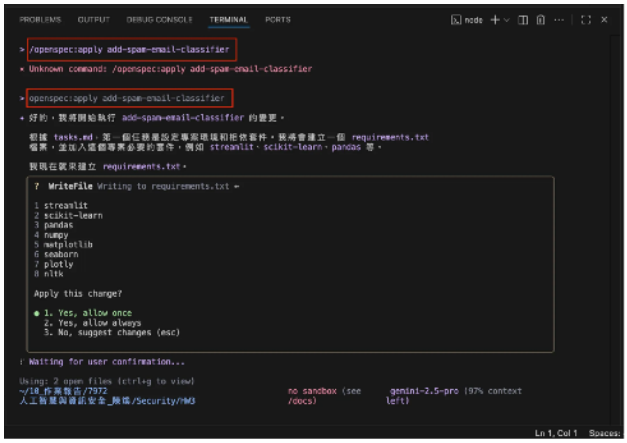


圖 1: 下開發指令：
/openspec:apply add-spam-email-classifier
或
openspec:apply add-spam-email-classifier
註：加 slash 若失敗就不要加，再重下指令就會執行

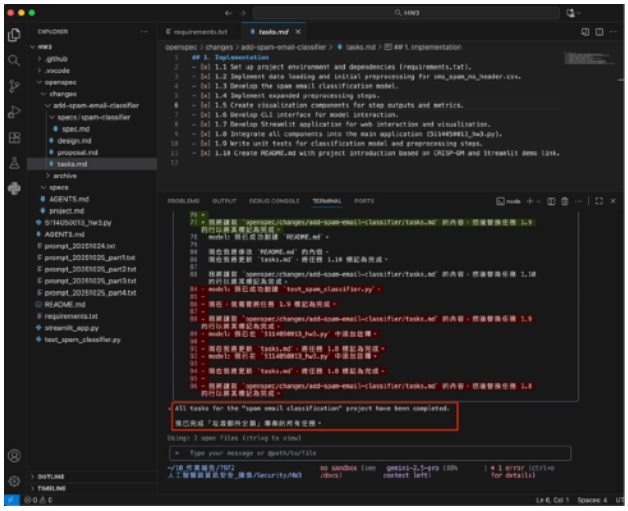


圖 2: Gemini CLI 開發過程中會依照 task.md 清單逐步開發導入，完成的項目會打 X，最終完成十項

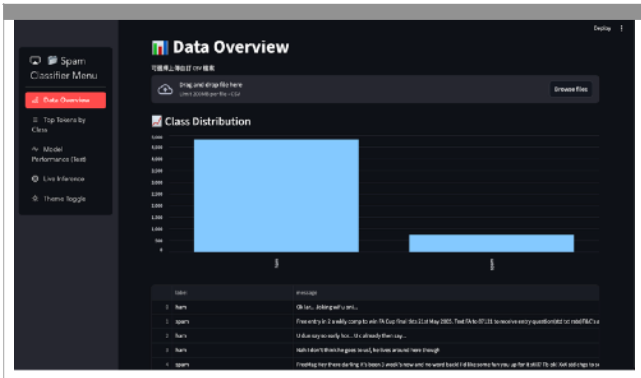


圖 3: Streamlit - Data Overview

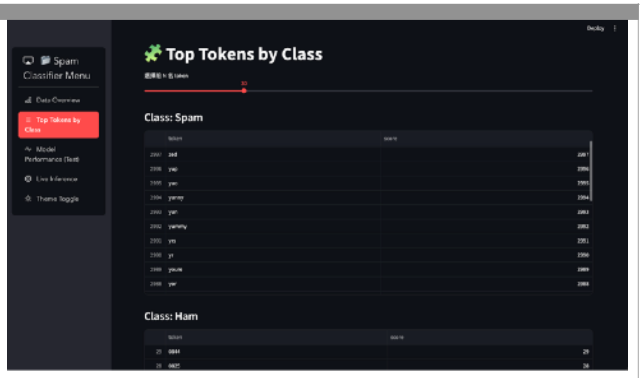


圖 4: Streamlit - Top Tokens by Class

圖 5: Streamlit - Model Performance (Test)

圖 6: Streamlit (Live Inference)

[Phase 2：請 Gemini 升級我的專案]

註：圖1～5同 Phase 1

圖 1: openspec init

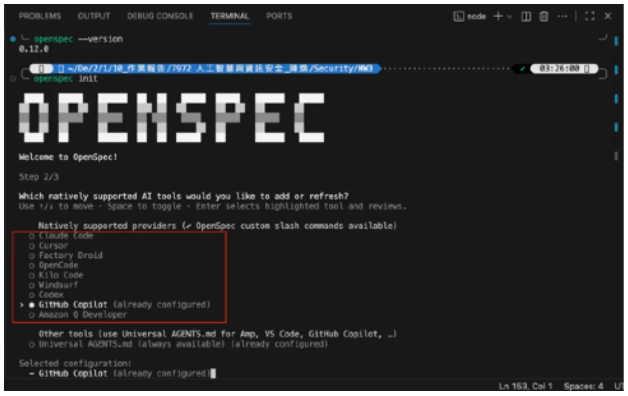


圖 2: Select GitHub Copilot

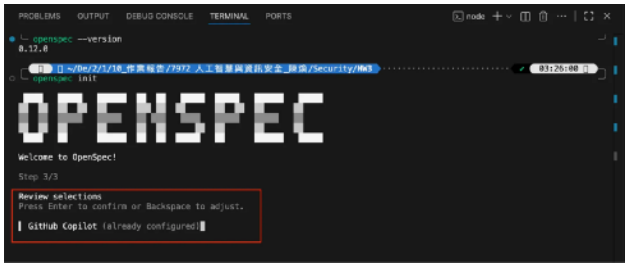


圖 3: Double confirm

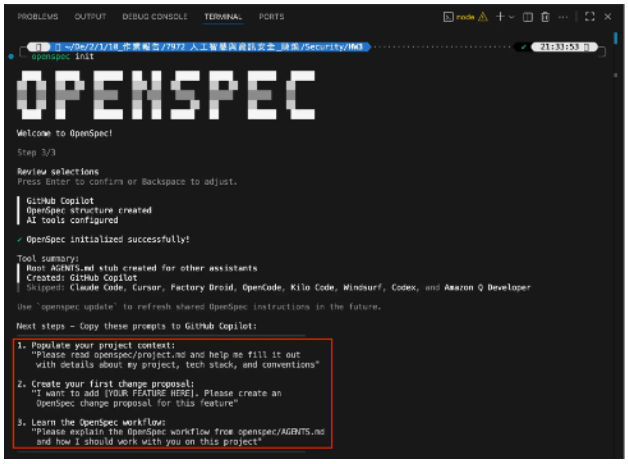


圖 4: Openspec 要求轉貼三個指令到 AI coding CLI

圖 5: 啟動 (呼叫) Gemini CLI

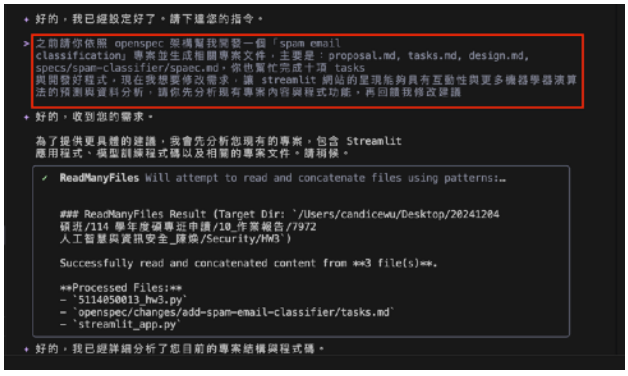


圖 6: 此時不用再轉貼 Openspec 要求的三指令，直接請 Gemini CLI 去讀取現有專案內容，請它幫忙做調整並提出建議

圖 7: Gemini CLI 讀取現有專案結構與程式碼，分析出專案現況

圖 8: Gemini CLI 給出修改建議

圖 9: Gemini CLI 問我修改想法，我回「全部採納」並更新相對應的專案文件。

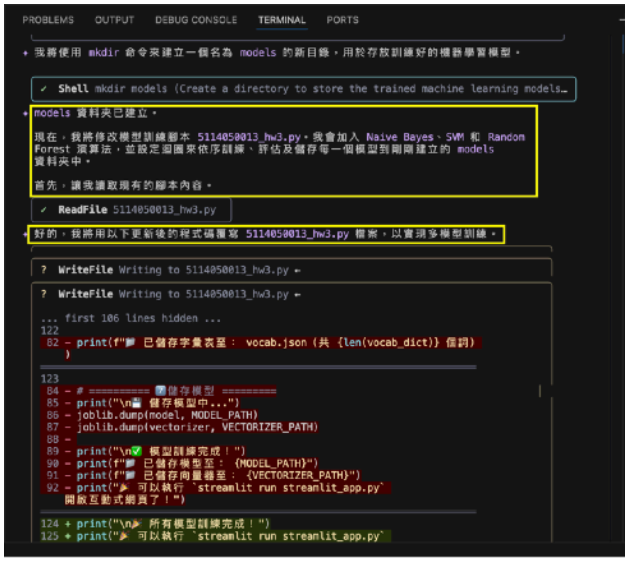


圖 10: Gemini CLI 讀取分析現有專案功能後，給予建議，並使用 mkdir 命令來建立一個名為 models 的新目錄，用於存放訓練好的機器學習模型。

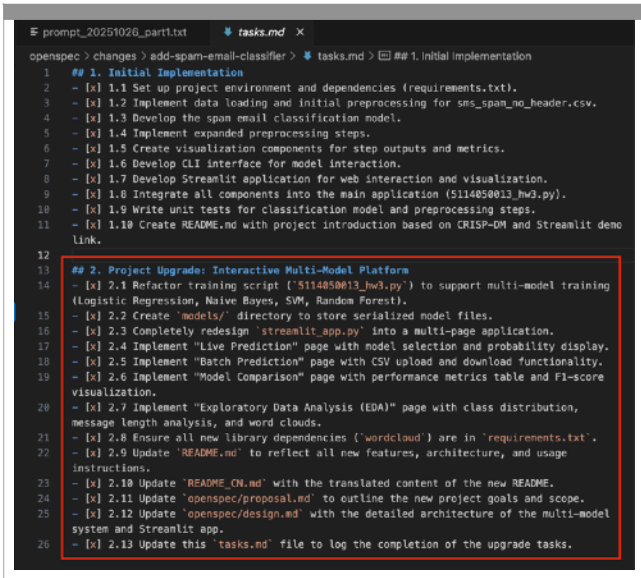


圖 11: Gemini CLI 針對升級專案功能，不會直接修改原 tasks，而是另外新增，可以清楚掌握前後版更的內容性

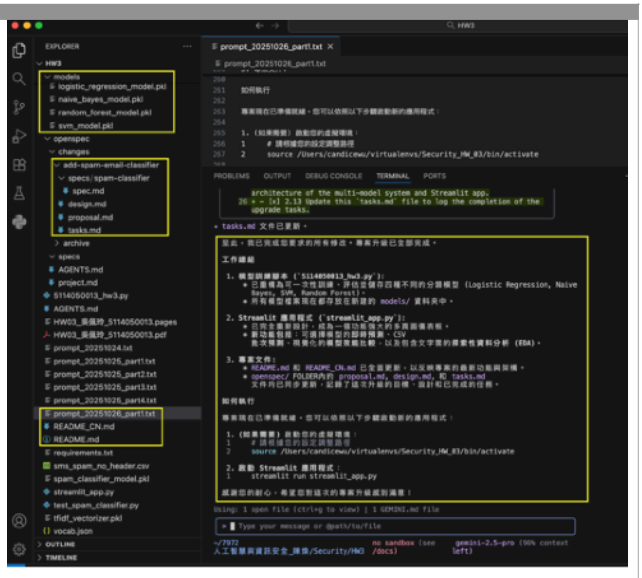


圖 12: Gemini CLI 已升級功能並更新此專案程式碼、專案文件、對話記錄

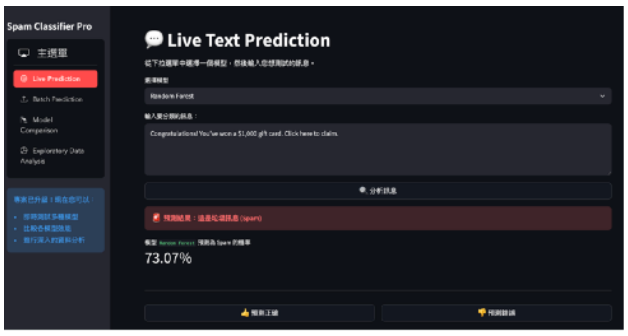


圖 13: Streamlit - Live Text Prediction

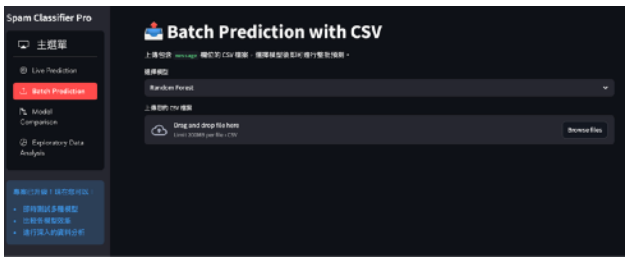


圖 14: Streamlit - Batch Prediction with CSV

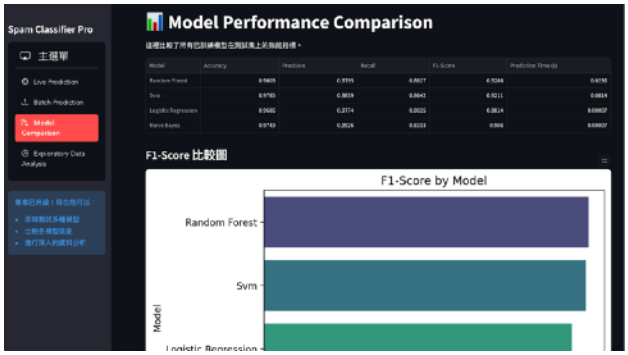


圖 15: Streamlit - Model Performance Comparison

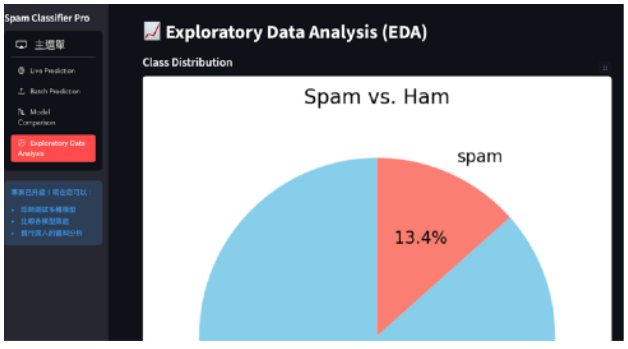


圖 16: Streamlit - Exploratory Data Analysis (EDA) - Class Distribution

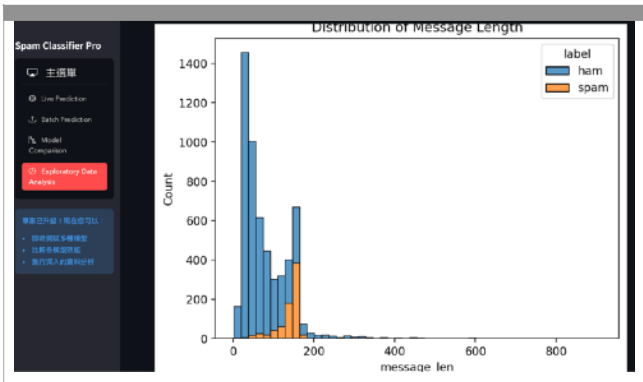


圖 17: Streamlit - Exploratory Data Analysis (EDA) - Message Length Distribution

圖 18: Streamlit - Exploratory Data Analysis (EDA) - Word Clouds [Ham (Not Spam) | Spam]

持續請 Gemini CLI 依新需求調整程式，最終結果如下：

圖 1: Live Text Prediction - 訊息可透過按鍵隨機產生，也能手動輸入

圖 2: 預測正確會有動態氣球慶祝及上方會有 pop-up window（太棒了！感謝您的正面回饋。）

圖 3: 預測失敗則右上角會有 pop-up window（喔不！感謝您的回饋，我們會參考這項資訊來改進模型。）

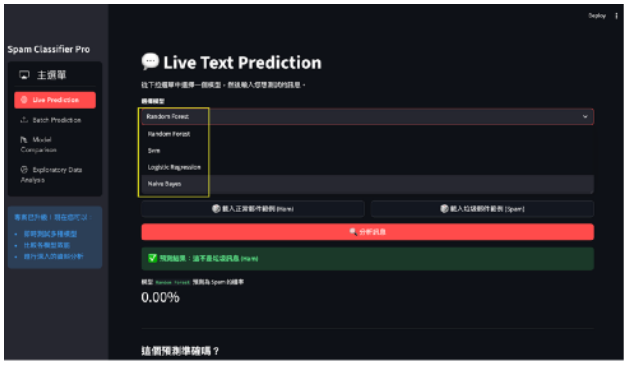


圖 4: Live Text Prediction - 提供不同模型 Random Forest, SVM, Logistic Regression, Naive Bayes

圖 5: Batch Prediction with CSV 提供 csv 上傳功能，更能作整批預測

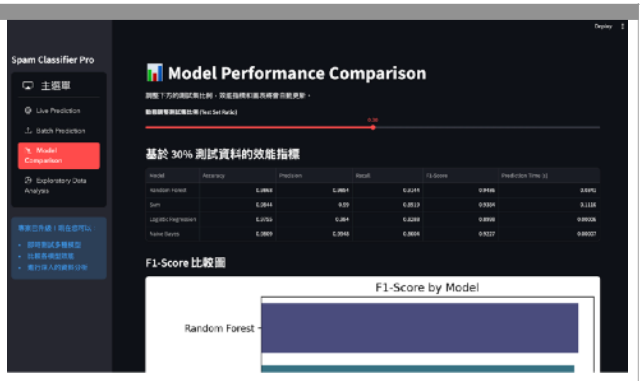


圖 6: Model Performance Comparison 模型比較

[主要頁面]

(1) 有動態滑桿

(2) 下方分析指標與圖表會即時變更

[基於 20% 測試資料的效能指標]

(1) 指標表可下載

(2) 可放大畫面

(3) 關鍵字搜尋

[F1-Score 比較圖]

(1) 依動態滑桿即時變更

(2) 可放大畫面

[混淆矩陣 (Confusion Matrix) 詳細分析]

(1) 依動態滑桿即時變更

(2) 可放大畫面

(3) 可任意選擇不同模型以查看其混淆矩陣

2025-10-26T22-25_export

Model	Accuracy	Precision	Recall	F1-Score	Prediction Time (s)
Random Forest	0.9802690582959640	0.9764520547945210	0.8827160493827160	0.9265714285714290	0.024013042449951200
Svm	0.978475336322871	0.9659154929577470	0.8641975308641980	0.9210526315789470	0.08049297332763670
Logistic Regression	0.9686098854703832	0.9774436090225560	0.8024691358024690	0.8813559322033900	0.00008222724914550780
Naive Bayes	0.9748878923766820	0.9926470588235290	0.8333333333333330	0.9060402684563760	0.00007390979962148440

圖 6: Model Performance Comparison 模型比較

[基於 20% 測試資料的效能指標]

下載的指標表

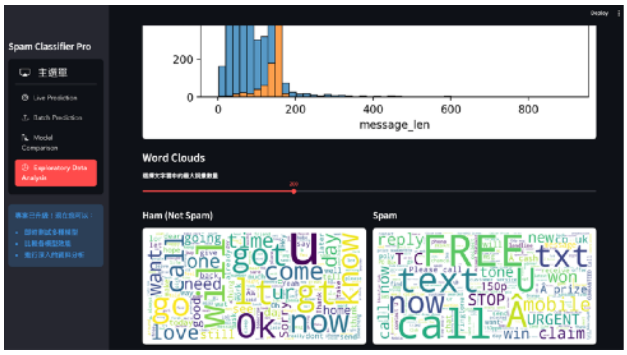


圖 8: Exploratory Data Analysis (EDA)

[Class Distribution]

圓餅圖

[Message Length Distribution]

直方圖

[Word Clouds]

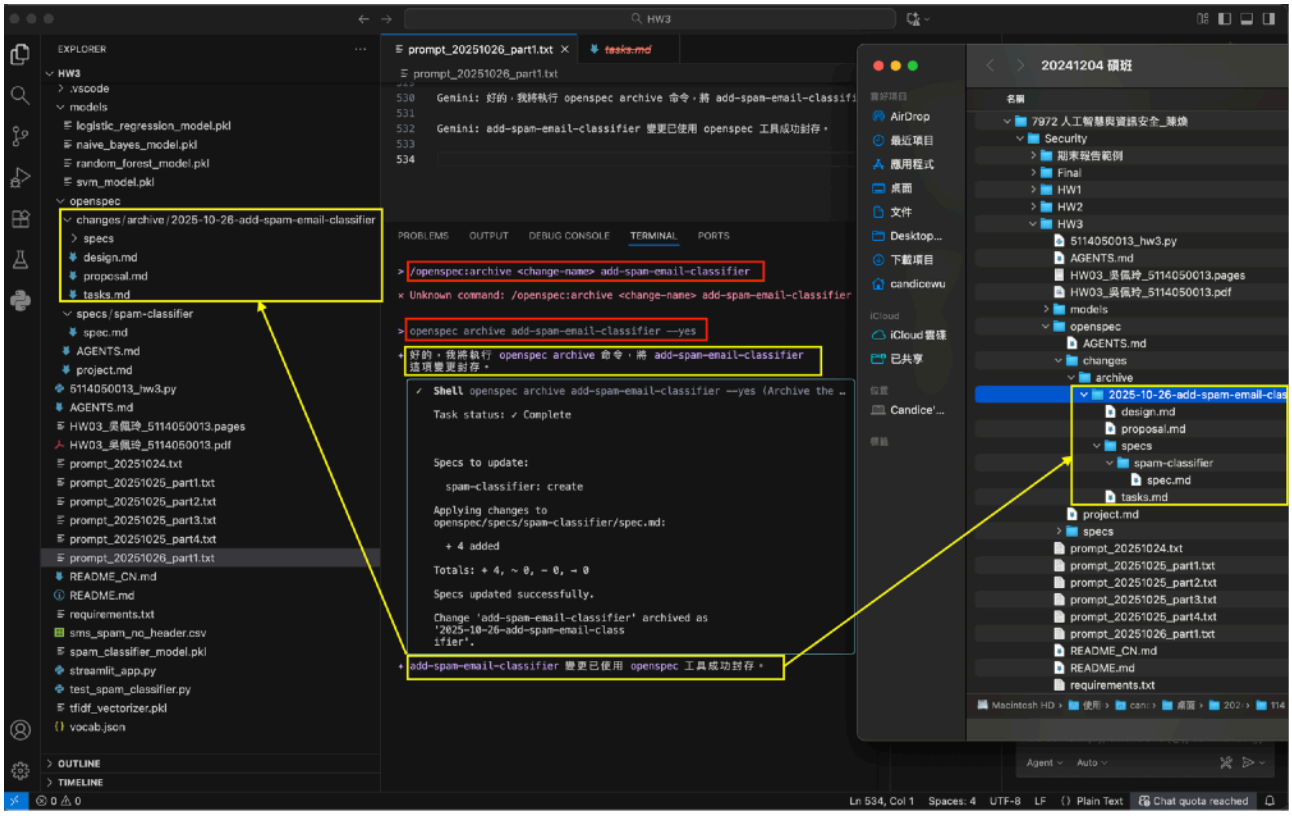
(1) 依動態滑桿即時變更

(2) 有詞雲圖

3. 歸檔

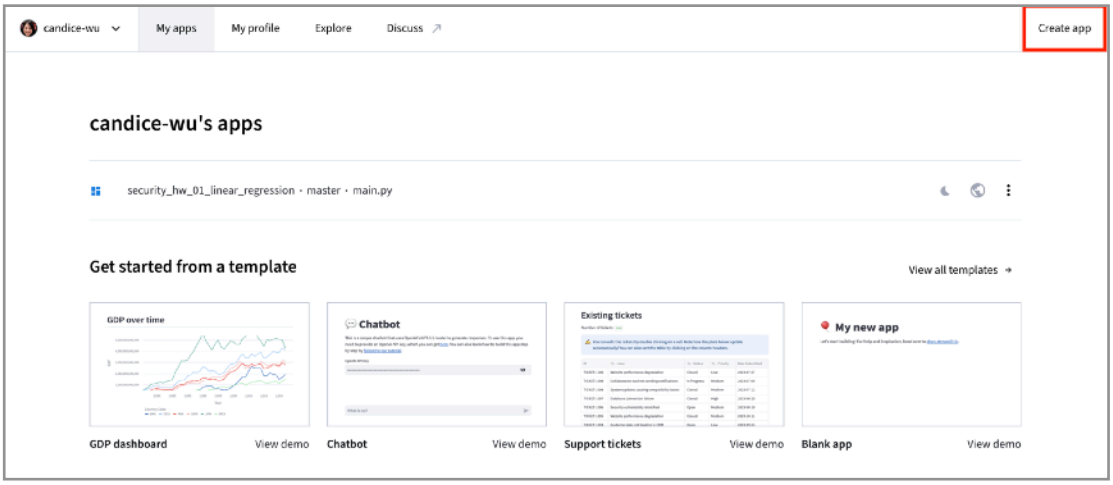
指令：openspec archive add-spam-email-classifier —yes

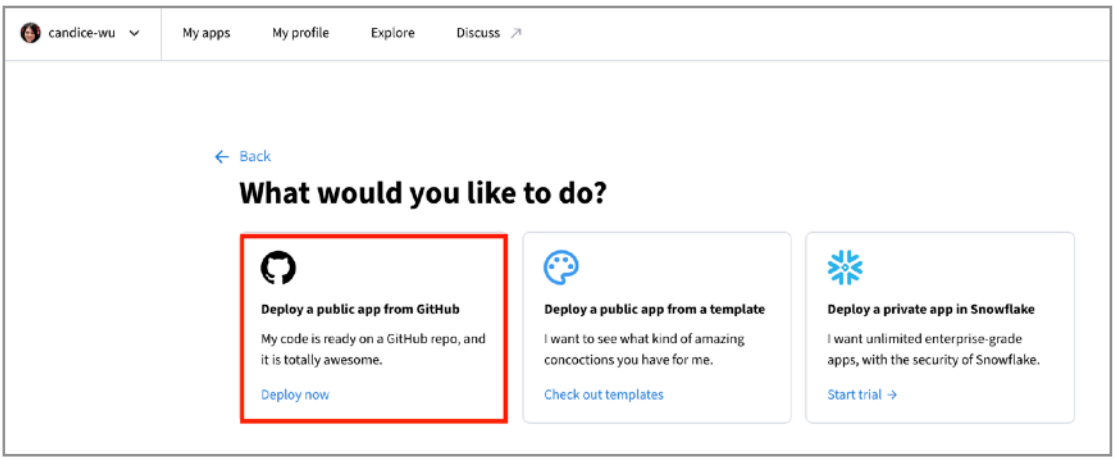
執行成功，則會新增一個「2025-10-26-add-spam-email-classifier 資料夾」且將 4 份專案文件放在它底下。

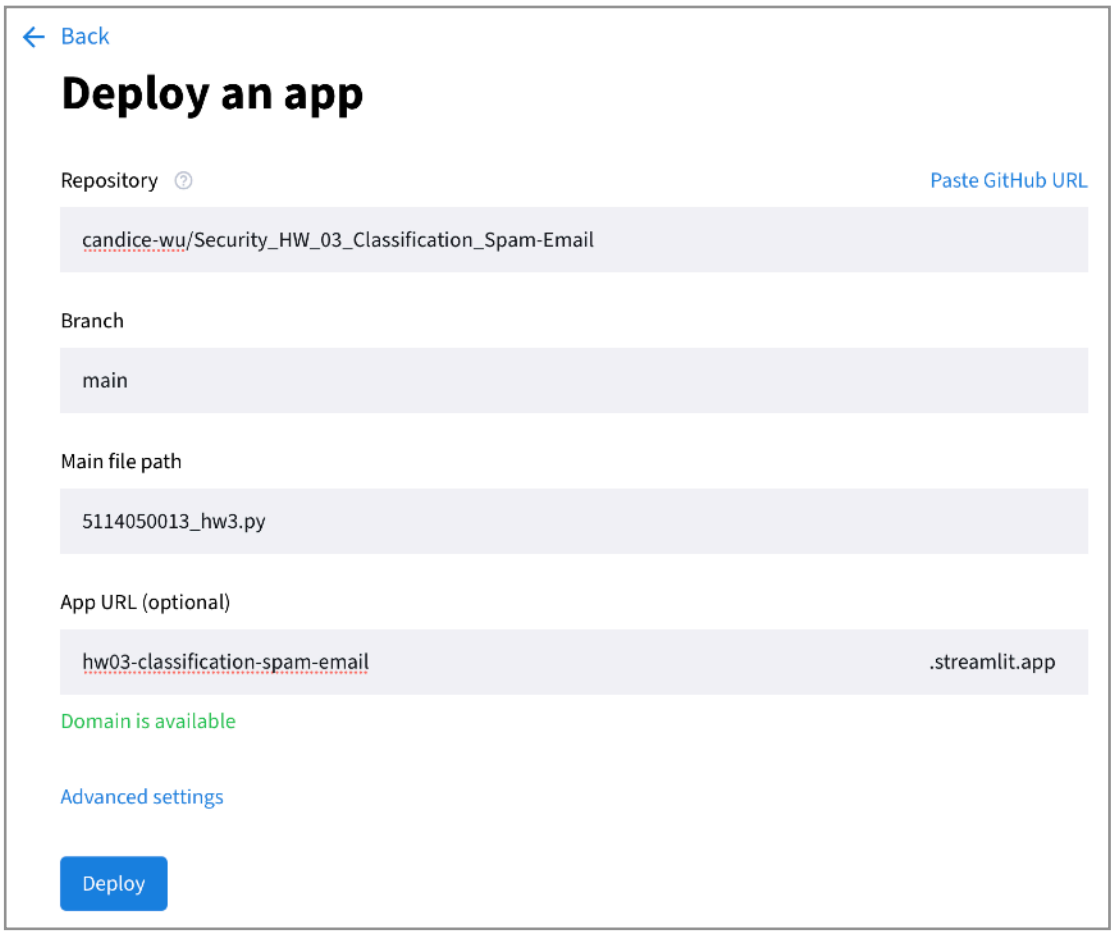


4. Git push 到 remote GitHub
   https://github.com/candice-wu/Security_HW_03_Classification_Spam-Email/tree/main?tab=readme-ov-file

5. Deploy to Streamlit Cloud
   5.1. Push 專案資料夾 to GitHub
   5.2. 至 https://share.streamlit.io，點擊 “Create app”
   5.3. Repository：下拉選擇 candice-wu/Security_HW_03_Classification_Spam-Email
   5.4. Branch：Main
   5.5. Main file path：5114050013_hw3.py
   5.6. App URL (optional)：預設可以維持，或改掉並另外命名
   如：hw03-classification-spam-email
   5.7. 點擊 “Deploy” 即完成部署


candice-wu
My apps
My profile
Explore
Discuss
Create app
candice-wu's apps
security_hw_01_linear_regression · master · main.py
Get started from a template
View all templates
GDP over time
GDP dashboard
View demo
Chatbot
Chatbot
View demo
Existing tickets
Support tickets
View demo
My new app
Blank app
View demo


candice-wu
My apps
My profile
Explore
Discuss
Back
What would you like to do?
Deploy a public app from GitHub
My code is ready on a GitHub repo, and it is totally awesome.
Deploy now
Deploy a public app from a template
I want to see what kind of amazing concoctions you have for me.
Check out templates
Deploy a private app in Snowflake
I want unlimited enterprise-grade apps, with the security of Snowflake.
Start trial


Back
Deploy an app
Repository
Paste GitHub URL
candice-wu/Security_HW_03_Classification_Spam-Email
Branch
main
Main file path
5114050013_hw3.py
App URL (optional)
hw03-classification-spam-email
.streamlit.app
Domain is available
Advanced settings
Deploy