期末專案名稱 (建議)：AI 美食指南：Google Maps 餐廳評論速讀與摘要

41147022S 謝宇宸

一、 使用模型介紹

本專案核心採用了 Google 開發的輕量級、開源大型語言模型 (LLM) google/gemma-7b-it。Gemma 模型系列基於與其更大型的 Gemini 模型相同的研究和技術建構，旨在提供高效能且易於研究人員和開發人員使用的選擇。選擇 gemma-7b-it (70 億參數指令微調版本) 的主要原因是它在理解指令、生成文本方面具有相對優良的表現，同時透過量化技術，有機會在 Google Colab 這樣資源相對有限的環境中運行。

為了在 Google Colab 的 GPU (NVIDIA T4) 環境中高效運行 gemma-7b-it，我採用了 4-bit 量化技術。這透過 Hugging Face transformers 函式庫合 bitsandbytes 實現，具體設定是使用 BitsAndBytesConfig 來配置 load_in_4bit=True、bnb_4bit_quant_type="nf4" 以及 bnb_4bit_compute_dtype=torch.bfloat16。這種方式能夠顯著降低模型的 VRAM 佔用 (從原始 float16 的約 14GB 降至約 5-6GB)，使得在 Colab T4 GPU (約 15GB VRAM) 上進行推論成為可能，同時盡可能保留模型的效能。

在專案初期，我也曾嘗試過參數規模更小的 google/gemma-2b-it 模型進行原型開發和測試，為後續升級到 7B 模型積累了經驗。

二、 專案目的

在資訊爆炸的時代，消費者在選擇餐廳前往往會參考網路上的顧客評論。然而，熱門餐廳的評論數量可能非常龐大，逐一閱讀費時費力。本專案旨在解決這一痛點，達成以下目的：

1. 提升決策效率： 開發一個工具，讓使用者能快速從大量 Google Maps 餐廳評論中獲取核心觀點和整體評價。
2. 自動化摘要生成： 利用大型語言模型的自然語言處理能力，自動將多條零散的顧客評論提煉成一段簡潔、易懂的摘要。

3. 便捷的使用體驗： 透過 Gradio 建構一個簡單直觀的 Web 使用者介面，方便使用者輸入餐廳名稱並獲取 AI 生成的摘要。
4. 技術實踐與學習： 整合外部 API (Google Maps Platform)、探索大型語言模型的應用、掌握模型量化技術，並在資源限制下完成專案部署。

本專案的目標使用者是那些希望在短時間內了解餐廳口碑，以便做出更明智用餐選擇的普通消費者，特別是針對像台北市大安區這樣餐廳密集、資訊豐富的區域。

三、 專案重點與過程

1. 系統架構與 Gradio 介面設計： 本專案的核心流程是：使用者透過 Gradio 介面輸入餐廳名稱，後端 Python 程式呼叫 Google Places API 獲取餐廳資訊和評論，接著將評論文本傳給 Gemma-7B 模型進行摘要生成，最後將結果回傳並在 Gradio 介面上展示。 介面設計力求簡潔，主要包含餐廳名稱輸入框、提交按鈕，以及用於展示餐廳基本資料、原始評論摘錄和 AI 評論總結的輸出區域。

2. Google Maps API 整合與評論獲取： 我使用 googlemaps Python 函式庫來串接 Google Places API。首先需要設定 API 金鑰，並啟用 Places API 服務。過程中遇到的挑戰包括：

- 初期遇到 REQUEST_DENIED 錯誤，經查閱文件發現是呼叫了舊版 API 端點，需確保專案啟用了正確的 Places API 版本。
- API 對於單次請求回傳的評論數量有限（通常最多 5 則「最相關」評論），這也限制了我 AI 分析的原始資料量。

3. 大型語言模型 (Gemma-7B) 的選擇、載入與量化： 選擇 gemma-7b-it 是為了在 Colab 環境下追求更好的生成效果。載入此模型需要：

- 在 Hugging Face 網站上同意 Gemma 模型的使用條款。
- 產生 Hugging Face Access Token，並在 Colab Secrets 中設定 HF_TOKEN 以供身份驗證。
- GPU 的重要性： 初期曾忘記在 Colab 中啟用 GPU，導致 gemma-2b-it 運行極其緩慢（單次請求長達 5 分鐘以上）。啟用 GPU (T4) 後，2B 模型的速度得到大幅改善，也為後續挑戰 7B 模型奠定了基礎。
- 量化探索歷程：

    - 嘗試無量化 gemma-7b-it (float16)： 即使在 T4 GPU 上，也因 VRAM 不足 (約需 14GB+，超出 T4 上限) 而無法直接載入，或 device_map="auto" 將部分模型放到 CPU 導致極慢。
    - 嘗試 8-bit 量化 gemma-7b-it： 模型權重約 7-8GB，可以載入 T4 GPU。但在實際進行推論（生成文本）時，由於激活值 (activations) 和 KV Cache 的 VRAM 消耗，仍然遇到了 CUDA out of memory 錯誤，尤其是在處理多條評論拼接成的較長輸入時。
    - 最終採用 4-bit 量化： 透過 BitsAndBytesConfig 設定 load_in_4bit=True、bnb_4bit_quant_type="nf4" 等參數，成功將 gemma-7b-it 的 VRAM 佔用降低到約 5-6GB，使其能在 Colab T4 GPU 上穩定運行並進行推論。

4. 提示工程 (Prompt Engineering) 的迭代與優化： 初期，即使模型成功載入，LLM 生成的摘要品質也不理想，例如只回傳無意義的符號 (-) 或重複部分輸入內容。 透過在程式碼中加入詳細的 print 除錯語句，我追蹤到：

- Google Maps API 回傳的評論內容是否正確傳入。
- LLM 接收到的完整提示詞 (prompt) 結構。
- LLM 未經後處理的原始輸出 (Full decoded result before processing)。
- 經過後處理 (result[len(prompt_text):].strip()) 後的最終輸出。

針對這些問題，我對提示詞進行了多次迭代優化，主要策略包括：

- 賦予明確角色： 例如「作為一個客觀的美食評論分析員...」。
- 清晰的任務指令： 詳細描述期望的輸出格式、內容要點（如包含正面與負面評價）、字數限制。
- 否定性約束： 指示模型不要做什麼（如「不要自行計算平均分數」）。
- 引導性的輸出標記： 在提示詞末尾加上「專業評論總結：」來引導模型開始生成。 經過優化後，模型生成摘要的相關性和品質得到了明顯改善。

5. Gradio 介面功能調整： 根據專案進度和 LLM 在多任務上的實際表現，決定將功能聚焦。原先構想的「AI 風格食記片段」和「AI 提取推薦菜色」功能，由於在有限時間和資源下，提示工程的調校難度較高，且初步效果不如「評論摘要」穩定，因此在最終版本中予以移除，使專案核心目標更為明確。相應地，gradio_interface 函式和 Gradio UI 元件也進行了簡化。

四、 專案最終成果

本專案成功開發出一個名為「AI 美食指南：Google Maps 餐廳評論速讀與摘要」的 Gradio Web 應用程式原型。其主要成果如下：

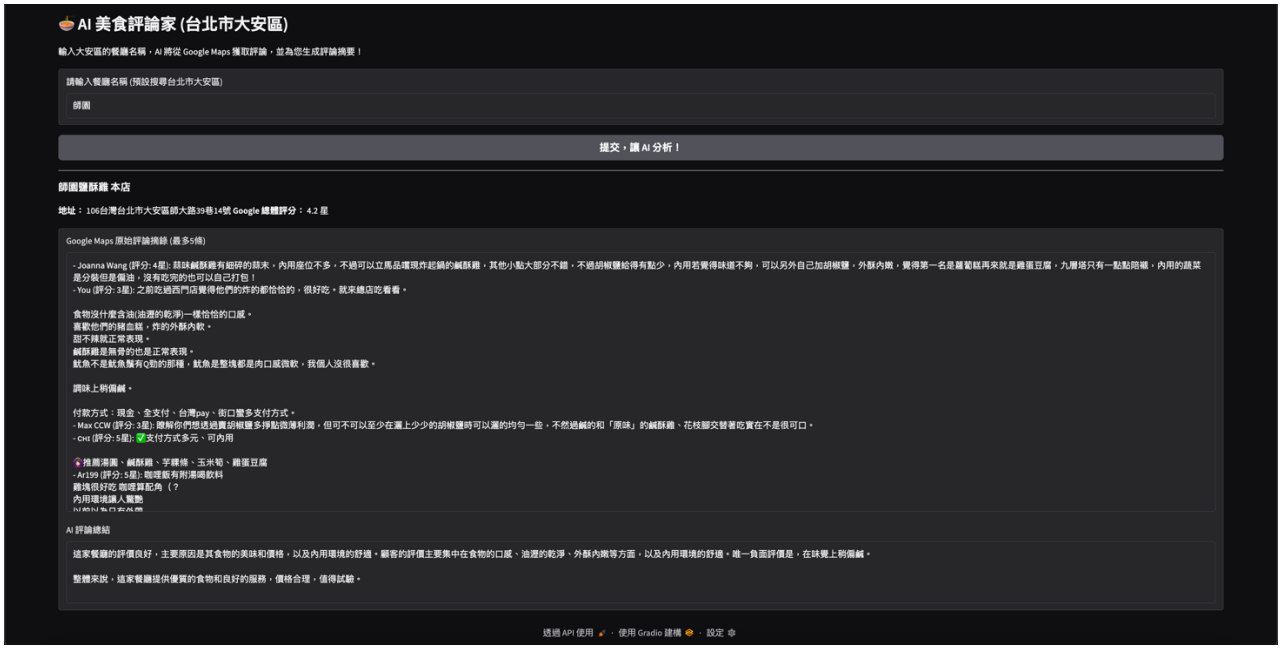
1. 功能實現：
    - 使用者可在介面輸入位於台北市大安區的餐廳名稱。
    - 應用程式能透過 Google Places API 自動獲取該餐廳的基本公開資訊（店名、地址、總體評分）以及最多 5 則顧客評論。
    - 利用 4-bit 量化的 google/gemma-7b-it 大型語言模型，對獲取到的評論進行分析，並生成一段約 50-80 字的「AI 評論總結」。
    - 所有資訊（餐廳資料、原始評論、AI 摘要）均清晰地展示在 Gradio 介面上。
2. 效能與環境：
    - 整個應用程式在 Google Colab GPU (T4) 環境下運行。
    - 透過模型量化和啟用 GPU，單次請求的處理時間（從提交餐廳名稱到顯示 AI 摘要）相較於初期 CPU 運行或未優化時有顯著縮短，達到了可接受的互動水平。
3. 學習與價值：
    - 本專案完整實踐了從數據獲取、AI 模型處理到使用者介面呈現的流程。
    - 深入學習了大型語言模型的載入、量化 (4-bit/8-bit)、提示工程以及在資源受限環境下的部署策略。
    - 掌握了 Gradio 建立互動式 AI 應用的方法。
    - 鍛鍊了 API 串接、錯誤處理和效能除錯的能力。
    - 最終成品可作為一個實用的工具原型，幫助使用者快速了解餐廳評價，提升生活效率。

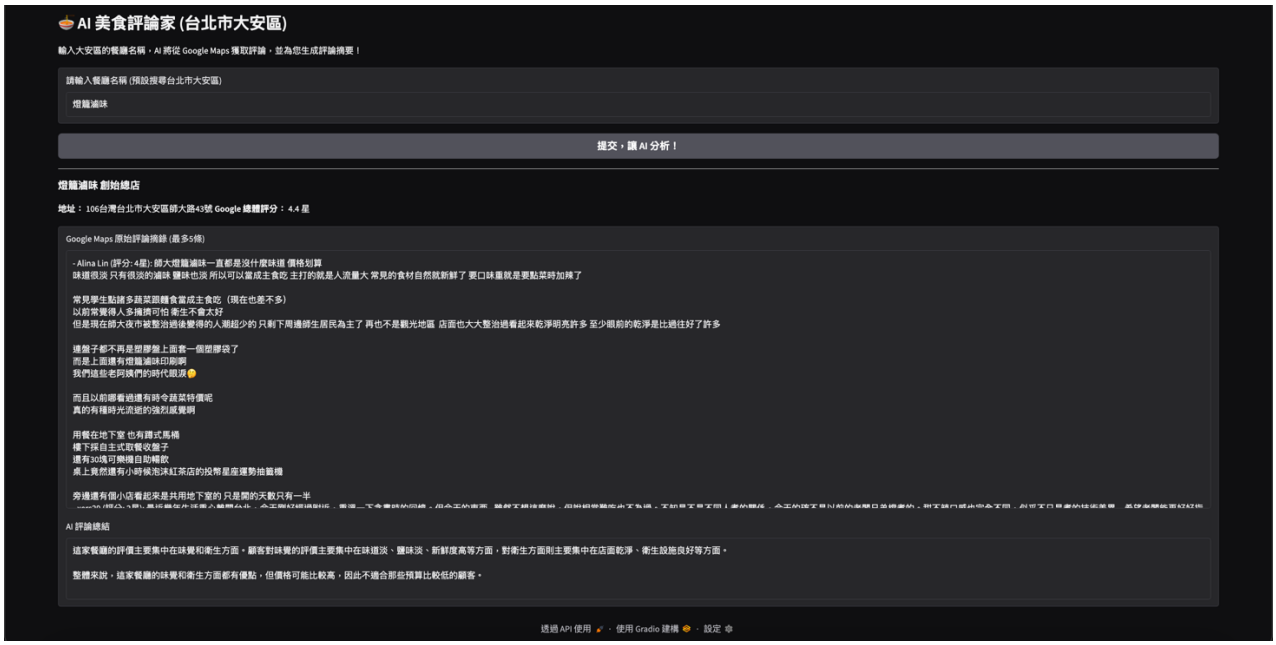
遇到的主要困難及解決方法回顧：

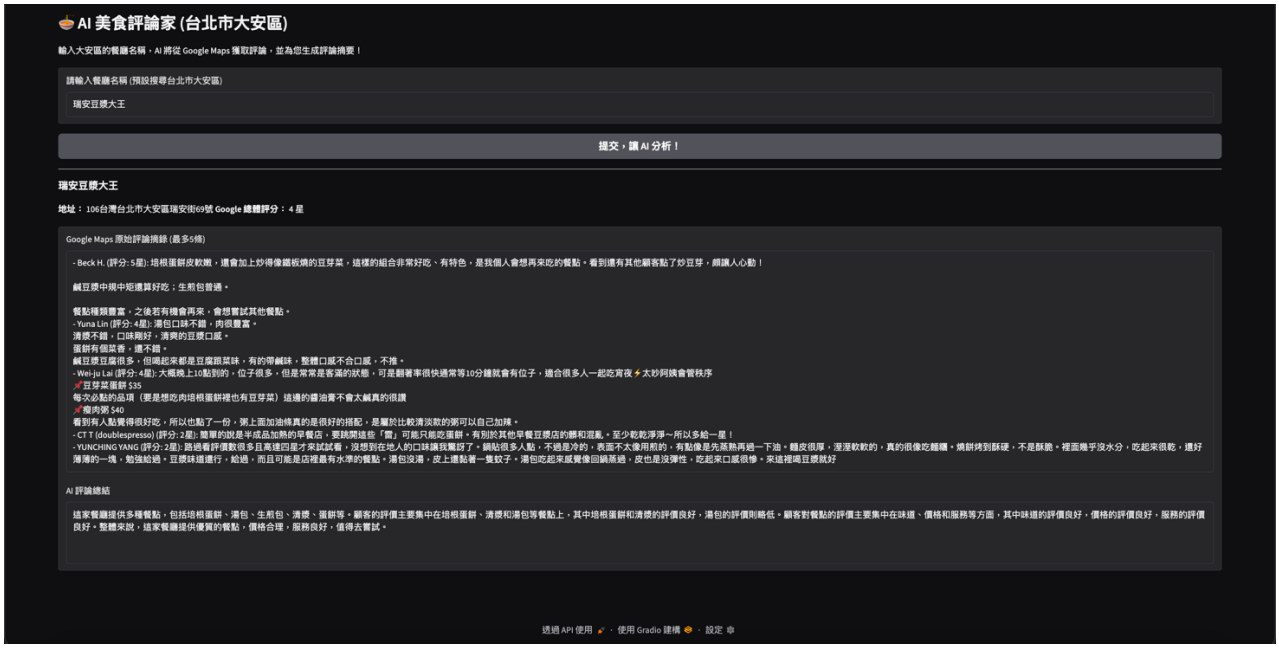
- API 權限與金鑰設定： 透過查閱官方文件逐步完成 Google Maps API 和 Hugging Face Token 的設定。
- 模型「Gated」存取： 在 Hugging Face 網站同意模型使用條款，並使用 Token 進行身份驗證。
- Colab GPU 未啟用： 初期因忘記啟用 GPU 導致運行極慢，後續修正並體會到 GPU 的重要性。

- VRAM 不足 (CUDA OOM)： 在嘗試 7B 模型時，從無量化、8-bit 量化到最終選擇 4-bit 量化，不斷調試以在 Colab T4 有限的 VRAM 中成功運行模型。
- LLM 輸出品質不佳： 大量投入時間進行提示工程的優化，透過明確指令、角色扮演等方式引導模型生成符合期望的內容。

截圖：







Google Colab 連結：
https://colab.research.google.com/drive/13CdvIF_CQENOvxDfOC7MS5e-wo0_MPU0?usp=sharing