威旭 x 清大期末 Kaggle 專題競賽：
高頻交易策略設計與研究

清華大學 計量財務金融學系大四

110071025 張智傑

一、模型差異化設計與亮點

相較於傳統僅以最後一筆 snapshot 特徵進行預測的模型設計，我採取了以序列為單位、結構為核心的建模邏輯。具體來說，我不單純仰賴單筆報價資訊，而是將整段 128 筆資料視為一個市場結構的縮影，透過時間序列統計與成交資訊交互，提取能反映市場趨勢、壓力與動能的複合特徵。

這樣的設計有幾個顯著特點：

- 區間統計視角：模型所使用的特徵來自整段資料的統計描述（如 mid_price、spread、imbalance、volatility），能反映短期內的市場變化，而非僅是單一時點的快照。
- 整合委託與成交資料：考慮了 VWAP、buy ratio、fill_density 等成交面變數，使模型同時捕捉「市場意圖」與「實際行為」。
- 以可解釋性為主軸的建模流程：從特徵設計到最終模型選擇，均強調模型的可解釋性與穩定性，並透過 SHAP 分析與特徵重要性檢視強化對模型邏輯的掌握。
- 避免過度複雜性：雖然嘗試過 ensemble 與 stacking，但最終選擇回歸單一 LightGBM 模型的原因，是其兼具穩定性與泛化能力，也最適合本任務中資料量有限、結構清晰的場景。

整體而言，我所設計的模型不僅強調預測準確性，也重視模型與市場行為之間的對應關係，希望能讓模型成為理解市場微結構的一種工具，而不只是單純的預測器。

那接下來，我會開始去說明起初自己對於競賽題目之理解、初期建構模型邏輯，然後會分成特徵篩選、模型選擇與調參去做過程思考之說明，同時補充自己在製作時不小心走入的思考盲區，最後統整與反思整體對於競賽的思維與體悟。

二、競賽題目理解與初期建構邏輯

本次競賽的核心目標是利用一段連續的委託與成交資料，預測第 128 筆委託在 10 分鐘後的價格。起初，我誤以為可以將任務視為一個靜態預測問題，因此僅針對最後一筆報價提取當下的價格、量與掛單資訊，直接餵入模型進行訓練。

然而這樣的設計雖在訓練集上表現良好，但在驗證與測試資料上出現明顯的準

確率崩壞。這使我意識到：單點 snapshot 的特徵遠不足以反映市場的動態變化，模型無法捕捉價格生成過程背後的結構性因素。

於是我重新檢視問題本質，將每組 128 筆資料視為一個短時間的「市場片段」，並思考：在這段時間內，市場是否出現價格趨勢？是否有明顯的買賣壓力？成交活躍與否？這些結構變化會如何影響未來價格？

基於這些觀察，我轉向設計具備統計意義與市場行為邏輯的特徵，例如：

- 價格水準與波動（如 mid_price_mean、log_return_std）
- 掛單深度與不對稱（如 depth_imbalance1、order_pressure）
- 成交活躍度與方向（如 buy_ratio、vol_buy_minus_sell）
- 報價與成交結構的偏離程度（如 mid_vs_vwap_avg）

這些轉變標誌著我從「靜態預測模型」走向「結構理解模型」的思維轉折，也為後續的建模與驗證奠定了正確方向。

三、特徵設計與篩選邏輯

特徵設計上，我以「價格趨勢、壓力、波動性與成交動能」為核心邏輯，期望能從整段市場片段中萃取出反映市場結構與行為變化的統計資訊，並建立具備預測力的特徵集合。整體特徵可分為以下幾類：

- 價格與波動性衡量：

mid_price_mean：報價中間價平均，代表市場價格中心水平。

spread_mean, spread_std：反映短期流動性與報價波動性。

log_return_std, rolling_std_mid_price：價格穩定程度與近期波動結構。

- 掛單深度與市場壓力：

bid_vol_total, ask_vol_total：觀察雙邊深度累積。

depth_imbalance1：第一檔掛單不對稱程度，評估主導方向。

order_pressure：觀察壓力波動性與是否反轉。

- 成交動能與交易主導性：

buy_ratio：買方成交比率，辨識是否主動買入。

vol_buy_minus_sell：買賣成交量差，量化多空力量強弱。

fill_density, avg_fill_speed：代表市場活躍度與交易頻率。

- 價格偏移與結構扭曲：

vwap_bid, vwap_ask：買/賣方成交價格重心。

mid_vs_vwap_avg：報價與實際成交價格間的差距，揭示市場偏誤程度。

為提升模型泛化性與穩定性，我進一步對上述特徵進行嚴謹篩選與優化，流程如下：

- 初步排序：使用 LightGBM 內建 feature_importance（Split 與 Gain）以及 SHAP value 排序作為初步依據，過濾掉極低重要度的特徵。
- 共線性分析：利用熱力圖計算皮爾森相關係數矩陣，將相關係數 $|r| > 0.85$ 的變數中刪除表現較弱者，降低多重共線性風險。
- 逐步刪除法（Ablation test）：每次刪除一項特徵，透過交叉驗證觀察 MSE 是否改善，以此方式確認是否有冗餘變數造成 noise。

SHAP 分析邏輯補充：

SHAP（SHapley Additive exPlanations）在此模型中扮演關鍵角色，它能提供：

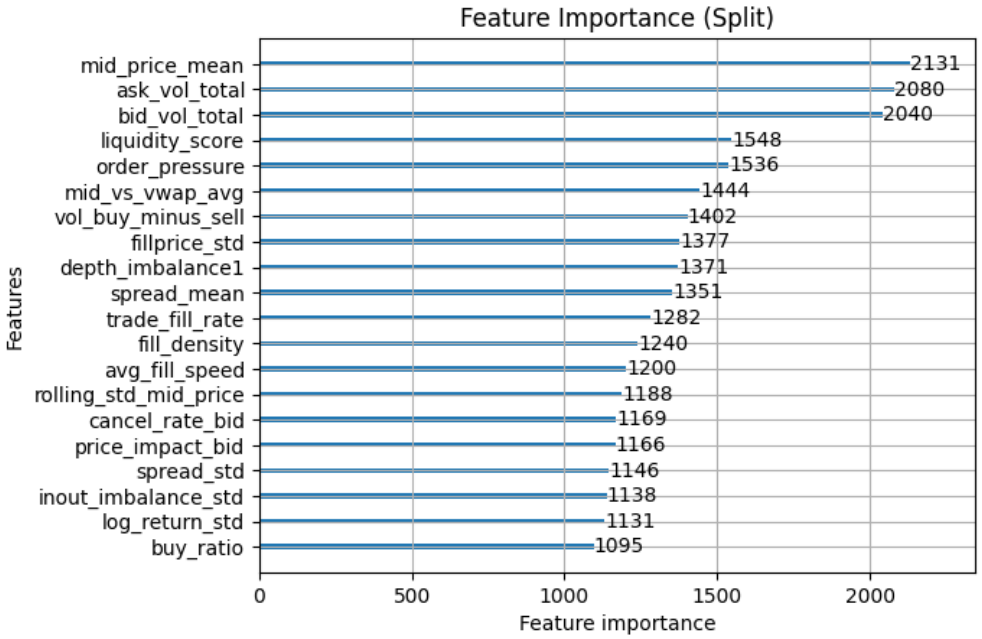
每個特徵對單一預測的邊際貢獻，了解所有特徵在整體樣本中的重要性分布，以及顯示特徵值變動對預測結果的方向與穩定性（如正向推升價格或壓抑價格）

透過 SHAP Summary Plot，我觀察到如 buy_ratio, vwap_bid, mid_price_mean 等對預測價格具有穩定貢獻，因此保留；而如 buy_streak_max, fillprice_vs_mid_diff 等雖直觀有意義，但 SHAP 值極低、變異大，實證貢獻有限，因此剃除。

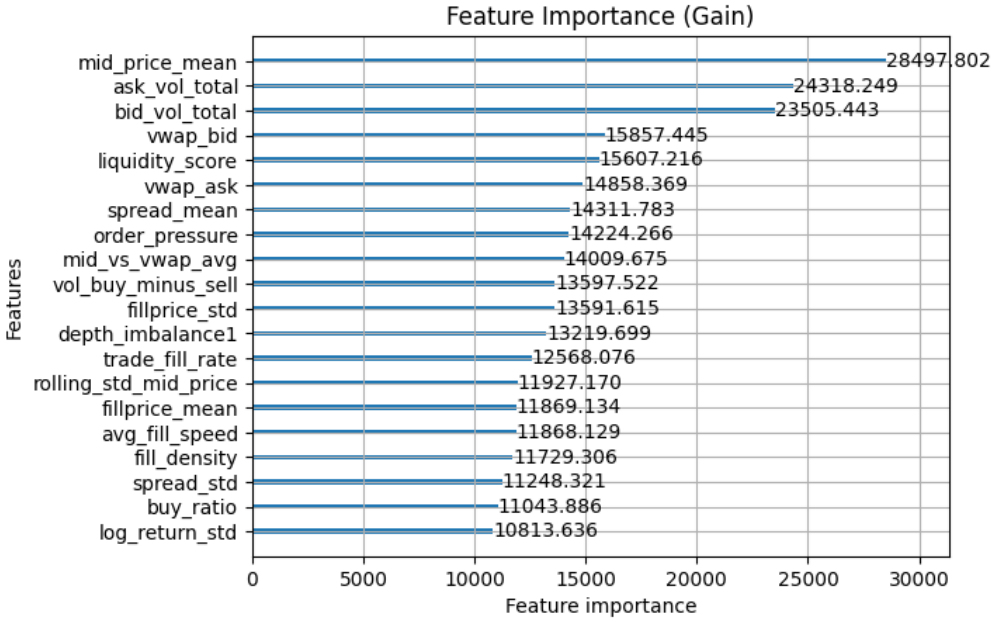
整體來說，我採取的是「基於市場邏輯設計 → 經驗與數據檢驗 → 精煉與去

雜訊」的流程，確保所保留的特徵能有效代表市場微結構與價格行為，並最終提升模型預測效能與穩定性。

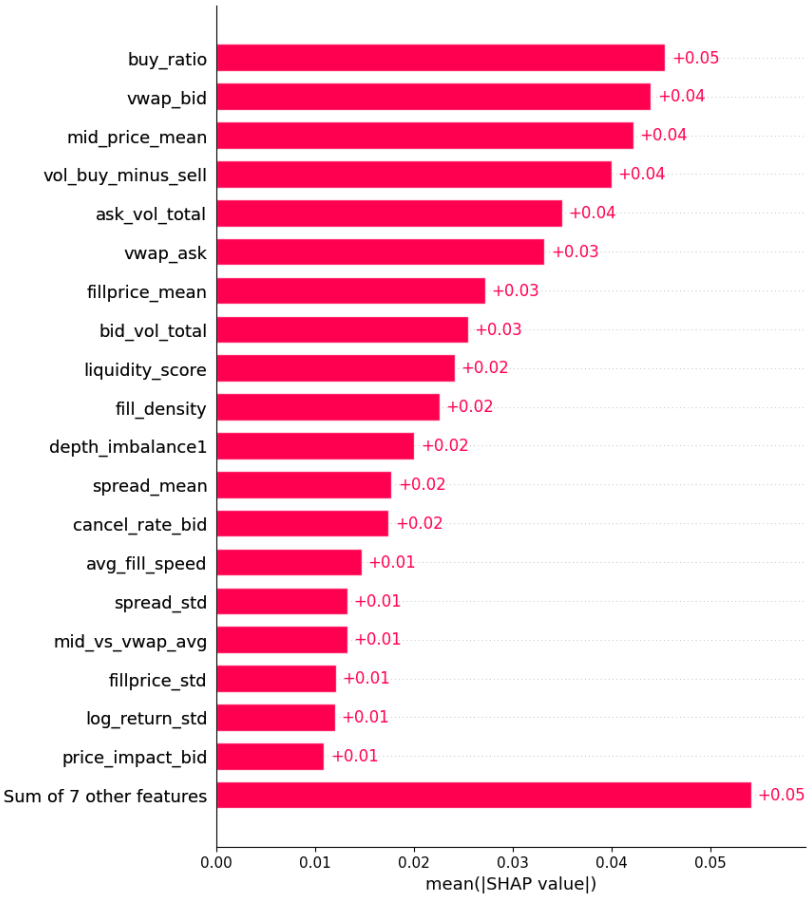
圖一：LightGBM 特徵重要性 (Split-based)



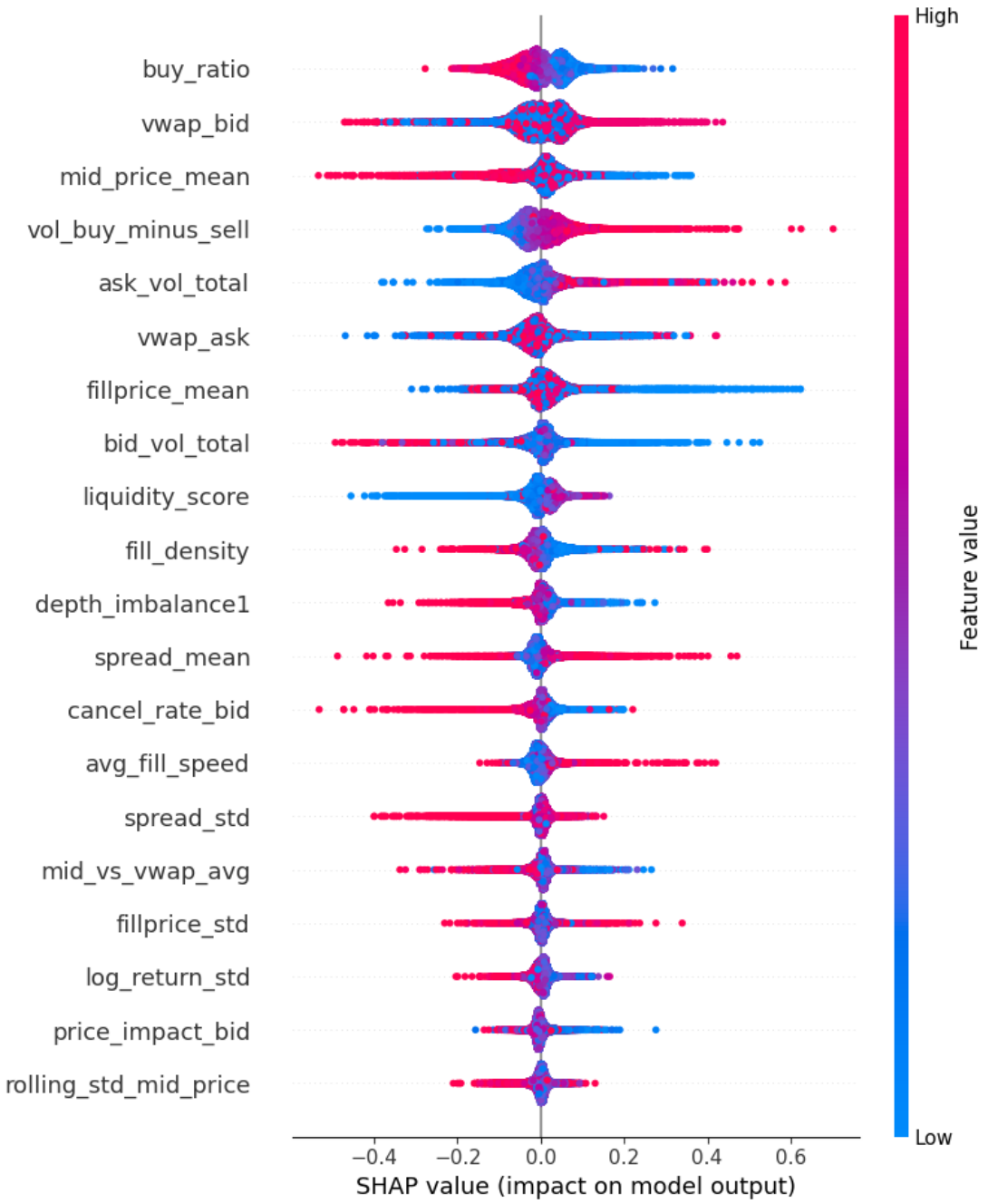
圖二：LightGBM 特徵重要性 (Gain-based)



圖三：SHAP 平均重要性條狀圖


buy_ratio +0.05
vwap_bid +0.04
mid_price_mean +0.04
vol_buy_minus_sell +0.04
ask_vol_total +0.04
vwap_ask +0.03
fillprice_mean +0.03
bid_vol_total +0.03
liquidity_score +0.02
fill_density +0.02
depth_imbalance1 +0.02
spread_mean +0.02
cancel_rate_bid +0.02
avg_fill_speed +0.01
spread_std +0.01
mid_vs_vwap_avg +0.01
fillprice_std +0.01
log_return_std +0.01
price_impact_bid +0.01
Sum of 7 other features +0.05
0.00
0.01
0.02
0.03
0.04
0.05
mean(|SHAP value|)

圖四：SHAP Value Beeawarm 解釋圖



四、模型選擇與參數調整

在模型選擇上，我最終選擇使用 LightGBM 作為主要預測架構。這項決定並不是一開始就確立的，而是自己經過多輪模型實驗、比較與思考後的結果。

1. 模型選擇邏輯與嘗試過程

在初期建模階段，我曾考慮多種模型結構，包括：

- XGBoost：表現穩定，但在本任務上與 LightGBM 的速度與 early stopping 整合性略遜一籌。
- Random Forest：泛化能力不錯，但缺乏增益導向的分裂邏輯，對於連續型數值特徵的學習表現不夠敏感。
- DART（Dropouts Meet Multiple Additive Regression Trees）：能抑制 overfitting，但在驗證集中表現浮動，泛化性不穩。
- Stacking（多模型融合）：雖提升了部分 fold 的表現，但推論速度慢，且在不同折數間 MSE 不夠穩定，最終捨棄。

在多輪驗證與交叉比較後，我發現 LightGBM 提供了最佳的穩定性與訓練效率，且結合 SHAP、特徵排序與 early stopping 等工具後，整體可控性強，非常適合我這次以「可解釋與穩健」為優先目標的建模思路。

2. LightGBM 的適用性分析

LightGBM 特別適用於本任務的原因如下：

- 高維特徵處理能力佳：本任務包含 20 多個數值型特徵，LightGBM 能自動選擇最佳切點並加速分裂。
- 支援 GroupKFold 與 early stopping：可準確找出最佳 boosting 輪數，防止過擬合。
- 模型可直接導出 Feature Importance 與 SHAP 整合解釋，利於模型剖析與後續精練。
- 訓練速度快，調參空間大：適合進行多輪 ablation 測試與 fine-tune。

3. 參數設計與調整策略

我採取保守偏穩健的參數配置原則，並配合 early stopping 找出最佳輪數。主要設計邏輯如下：

A. objective = regression：明確定義為連續變數預測問題，對應競賽的目標設定。
B. metric = mse：以最小化均方誤差為目標，與最終評分標準一致。

C. learning_rate = 0.05：控制每輪更新幅度，避免模型陷入不穩定震盪，提升收斂穩定性。
D. num_leaves = 31：限制每棵樹的最大複雜度，平衡模型能力與泛化風險。
E. feature_fraction = 0.9：每棵樹隨機選取 90% 特徵，增加多樣性，避免過度依賴特定特徵。
F. bagging_fraction = 0.9：每棵樹僅使用 90% 訓練資料訓練，有助於降低 overfitting。
G. bagging_freq = 5：每 5 輪重新抽樣一次資料，提供模型更多泛化空間。
H. seed = 42：統一隨機種子，確保結果一致性與可重現性。
I. verbosity = -1：抑制冗長輸出，保持訓練畫面簡潔。

透過這組參數配置與 GroupKFold（以 GroupID 分群）的 5 折交叉驗證，我能夠有效找出每輪的最佳訓練迭代數（通常介於 100~130 輪之間），並在 early stopping 機制下避免過度訓練。最終模型 retrain 時便可穩定使用該輪數作為完整訓練的依據，達成更佳泛化效果與穩定預測表現。

五、結語：學習歷程與反思

回顧本次競賽的建模過程，我認為這不僅是一場技術挑戰，更是一段思維修正與邏輯重塑的過程。

在最初的建模階段，我誤將此任務視為靜態預測問題，過度依賴第 127 筆報價的當下特徵。雖然模型在訓練集上表現出色，卻在驗證與測試集上快速失效，出現明顯的 overfitting。這讓我意識到，單點式的建模思維無法反映市場的動態性與結構性，也無法捕捉價格變化背後的真正邏輯。

於是我重新審視資料本質，將每組 128 筆資料視為具有結構意義的時間段，並設計出能夠反映價格趨勢、交易壓力與成交動能的統計特徵。這樣的邏輯轉換使模型不再只是「擬合過去的數字」，而是試圖理解資料背後的市場行為機制。

然而，模型調整進入優化階段後，我再度犯了一個常見錯誤：將過多心力投注在壓低訓練集的 MSE 上，忽略模型對未知樣本的穩定性與解釋能力。這是我第二次面對 overfitting，也促使我重新思考——一個好的模型，不是只能在已知資料表現優秀，而是必須在邏輯上能夠自洽、在未知資料上也能穩定發揮。

這次專案也讓我深刻學會如何在技術與邏輯之間取得平衡。LightGBM 雖不是最複雜的架構，但其高效率與可解釋性，反而讓我能更深入掌握特徵與預測間的關聯，並進行有效的調參與調整。模型效能不應僅止於指標最小化，更應建立在對資料與市場理解之上，才能從「數據擬合者」晉升為「結構解釋者」。

最終，這次競賽不僅提升了我在特徵工程、模型訓練與錯誤修正方面的能力，也讓我體會到資料科學的本質，是一個持續追問「為什麼這樣設計」的過程。我相信這樣的反思能力與架構思維，將成為我未來面對實務挑戰與研究探索時，最關鍵的底氣。