機器學習 期末報告
113352019黃得晉

1. Data: Sample period, predictors definition, etc.

從 TEJ 上下載台灣所有上市公司的資料，時間是從 2005 年 1 月開始，到 2025 年 3 月，會選擇這個時間段是因為我想把 2008 年的金融危機時期也考慮進去，那當初作業一開始在準備資料時，是在 2025 年 3 月的時候，所以時間才會只到 3 月，不然我覺得有把今年 4 月的資料有考慮進去的話，那結果應該會有很大的不一樣。我一開始下載的資料有

”證券代碼”, “年月”,” TEJ 產業_代碼”,” TEJ 產業_名稱”,” 員工人數-母公司”,” 統一編號”,” 設立日期”,” 首次掛牌日期”,” 收盤價(元)_月”,” 報酬率%_月”,” 週轉率%_月”,’” 流通在外股數(千股)”,” 市值(百萬元)”,” 本益比-TEJ”,” 股價淨值比-TEJ”,” 股價營收比-TEJ”,” 現金股利率”,” 該月結束日”,” 市場別”,” 成交筆數(千筆)”,” 成交值比重%”,” 成交量(百萬股)_月”,” 成交值(百萬元)_月”,” 市值比重%”,” 高低價差%”,”股價漲跌(元)”

接下來我的篩選條件是要有超過 11 年的資料且 2025 年 3 月還有資料的，11 年相當於 132 個月，所以我程式是寫要有 135 筆以上的資料長度，然後踢出我想考慮的變數中資料有超過連續三期為 0 的公司，最後將剩下的資料為 Nan 的話，則用下一筆的資料當作這一期的資料，最後留下來符合條件的公司數有 118 家公司。

我最後實際有考慮的變數是以下 12 個：

” 收盤價(元)_月”,” 週轉率%_月”,’” 流通在外股數(千股)”,” 市值(百萬元)”,” 股價淨值比-TEJ”,” 股價營收比-TEJ”,” 成交值比重%”,” 成交量(百萬股)_月”,” 成交值(百萬元)_月”,” 市值比重%”,” 高低價差%”,”股價漲跌(元)”

因為有些下載下來的變數並不是數值又或者說有些數值基本上不會變化，所以還是討論有明顯變動的資料來當作變數比較適當。

我有去計算各個公司個別變數的數據統計量，但資料太過龐大，所以我就只在程式碼裡呈現，下面圖表則是呈現所有數據的各個變數統計量

	mean	std	min	max	median
收盤價(元)_月	58.67	208.22	1.56	4764.06	21.61
週轉率%_月	10.92	18.20	0.0067	509.31	5.16
流通在外股數(千股)	2.97e+06	4.62e+06	2.35e+04	2.64e+07	6.35e+05
市值(百萬元)	1.73e+05	8.66e+05	3.36e+02	2.94e+07	2.54e+04
股價淨值比-TEJ	1.93	1.60	0	21.04	1.40
股價營收比-TEJ	2.56	10.65	0	657.77	1.46
成交值比重%	0.39	1.06	0	28.06	0.085
成交量(百萬股)_月	209.24	472.35	0	12417.00	62.00
成交值(百萬元)_月	1.08e+04	3.93e+04	0	1.59e+06	2.36e+03
市值比重%	0.57	1.89	0.001	39.08	0.095
高低價差%	13.12	9.50	0.61	156.86	10.78
股價漲跌(元)	0.36	25.98	-1120.00	660.00	0.03
報酬率%_月	1.12	9.93	-47.27	144.28	0.40

其實這個表好像不太能看出什麼，因為各個公司去綜合比較的話，那有些資料的全距就會很大，這也有可能導致標準差很大，那看似很分散的資料，再以單一公司來看時，變化不大標準差很小，那實際上在運用時，效果就會很差。

2. Methodology: Linear and non-linear models, what is the tuned hyper-parameter?

我 linear model 是選 Ridge，non-linear model 是選 XGboost，所有的隨機變數種子都選 31073。這次的結果都會跟之前所做的 Part1 有所比較，所以當我提到 Part1 時，參數的部分就是用上一份報告所提到的參數，下面等等也會仔細提到，而這次所使用的參數，後面的敘述都會用新參數來表示。所以請注意 Part1 跟新參數的用詞，來區分圖表結果。

1. Ridge model 選擇的超參數：
   - alpha：[0.01, 0.05, 0.1, 0.5, 1.0, 5.0, 10.0, 20.0, 30.0, 40.0, 50.0]，這是用於正則化強度的候選值，上次 Part1 只有[0.01, 0.05, 0.1, 0.5, 1.0, 5.0, 10.0]，但因為算出來的最佳參數都是 10，所以我這次提高了可選的 alpha，看看效果會不會比較好，自己預期希望結果是不要發生在 0.01 跟 10.0，因為在邊界可能會有訓練不完全的問題。
   - store_cv_results：這裡我是有打開儲存交叉驗證的結果。

2. Ridge model 未選擇的超參數（所以是使用 scikit-learn 的預設值）：
   - fit_intercept：預設為 True，表示模型會擬合截距。
   - scoring：預設為 None，表示使用模型用 mean squared error 的評分方法。
   - cv：預設為 None，代表用 5 折交叉驗證。
   - gcv_mode：預設為 None，代表使用廣義交叉驗證（Generalized Cross-Validation, GCV）來選擇最佳 的 alpha。

3. XGBoost model 選擇的超參數：
   - max_depth：[3, 5, 7, 9, 11]，上次 Part1 是選 [3, 4, 5, 6, 7]，但結果都是發生在=7 的時候，因此這次才選改選擇[3, 5, 7, 9, 11]，自己是希望結果不要發生在 3 跟 11 的地方，因為在邊界可能會有訓練不完全的問題。
   - learning_rate：[0.01, 0.1, 1.0]，上次 Part1 是選[0.01, 0.05, 0.1, 0.5, 1.0]，可結果是 0.1，因為其他參數可選擇的數字這次會增加，所以就想說這次在這部分減少可選的參數，來降低訓練的時間，不然我擔心訓練時間會過長。
   - n_estimators： [100, 200, 300, 400, 500]，上次 Part1 是選[100, 200, 300]，但因為上次訓練完的結果都是選 300，所以這次才增加 400 跟 500 的選項，自己是希望結果不要發生在 100 跟 500 的地方，因為在邊界可能會有訓練不完全的問題。
   - objective：固定設定為 squarederror，代表使用大家比較常用的均方誤差作為損失函數。

4. XGBoost model 未選擇的超參數：

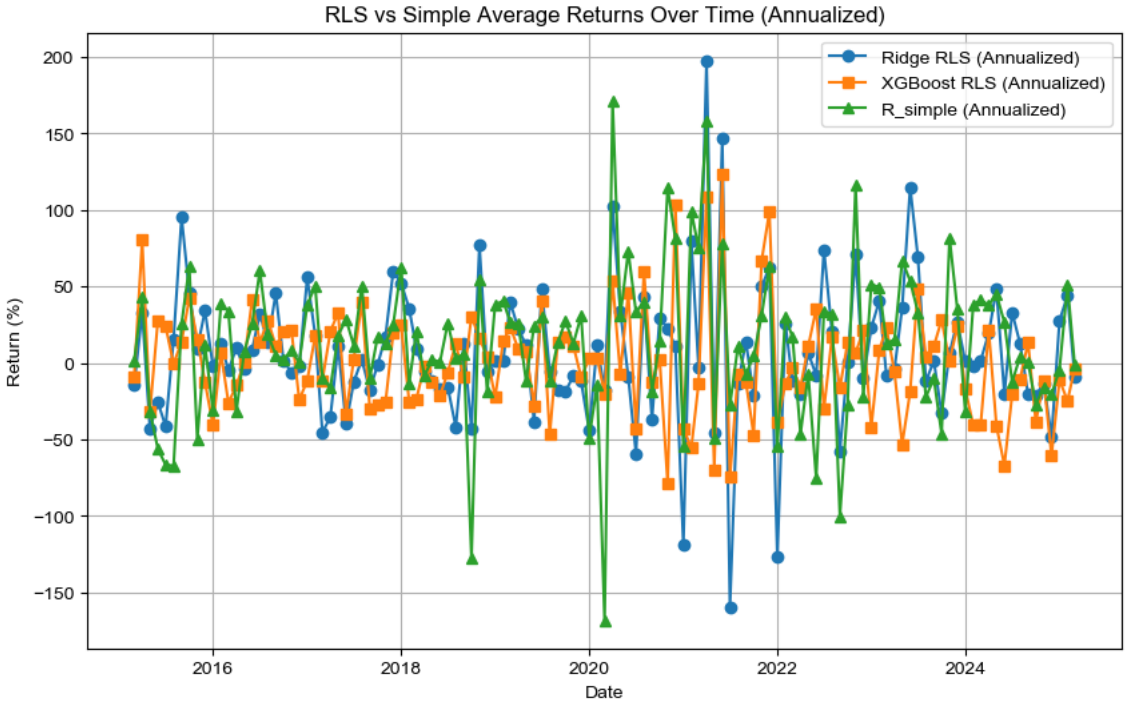
- subsample：預設為 1.0，表示使用所有訓練數據進行每棵樹的訓練，未進行子採樣。
- colsample_bytree：預設為 1.0，表示每棵樹使用所有特徵，未進行特徵子採樣。
- colsample_bylevel：預設為 1.0，表示每層分裂時使用所有特徵。
- colsample_bynode：預設為 1.0，表示每個節點分裂時使用所有特徵。
- gamma：預設為 0，表示不對樹的分裂設置最小損失減少要求。
- min_child_weight：預設為 1，表示葉子節點的最小權重，控制過擬合。
- reg_alpha：預設為 0，表示無 L1 正則化（Lasso）。
- reg_lambda：預設為 1.0，表示 L2 正則化（Ridge）的強度。
- max_delta_step：預設為 0，通常在平衡數據集中無需調整。

結果訓練總時長：約 125 分鐘，時間上大概是上一次 Part1 的兩倍。

3. Results

(a) The portfolio results: Do they beat the benchmark? What are the possible explanation for the good/bad performance?

先附上 Ridge RLS、XGBoost RLS 跟 benchmark(R_simple) 的年化報酬率圖表

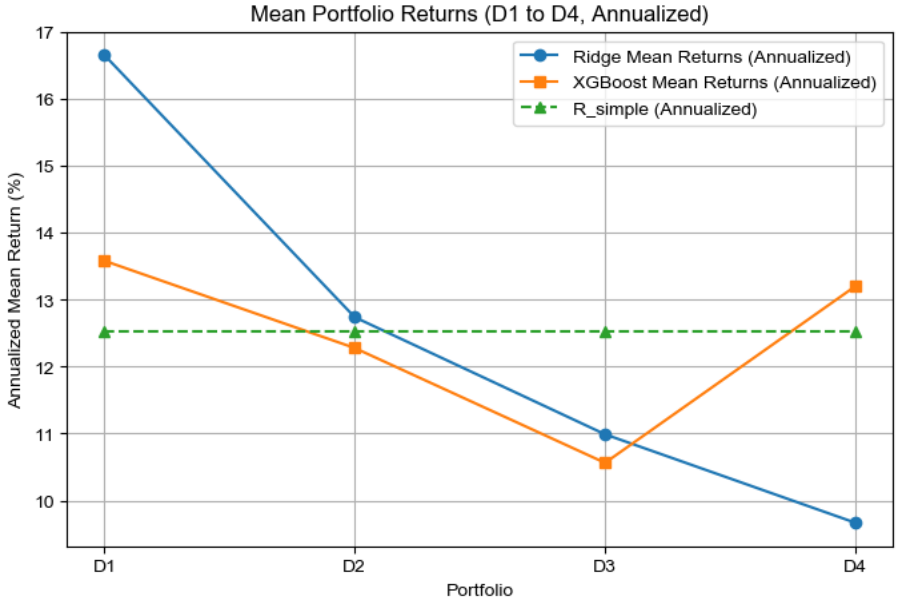


（上圖）這次新參數算出來的 RLS 跟 benchmark 的平均報酬

然後我分別去計算他們的 mean、Volatility、Mean-to-Volatility Ratio。

	Ridge	XGBoost	Benchmark
Mean	6.990449%	0.379973%	12.5243%
Volatility	46.017315%	35.964371%	48.1598%
Mean-to-Volatility Ratio	0.151909	0.010565	0.2601

雖然說 Volatility 是有比 benchmark 小，但平均報酬跟 Mean-to-Volatility Ratio 整體上來說，還是差距蠻大的，所以我自己會覺得說我訓練出來的 RLS 投資組合不論是用線性還是非線性的結果，都是沒有打敗 benchmark 的。那可能出現的原因我們要先看我計算出來的 D1-D4 圖。



（上圖）這次新參數算出來的 D1-D4 圖

Ridge：

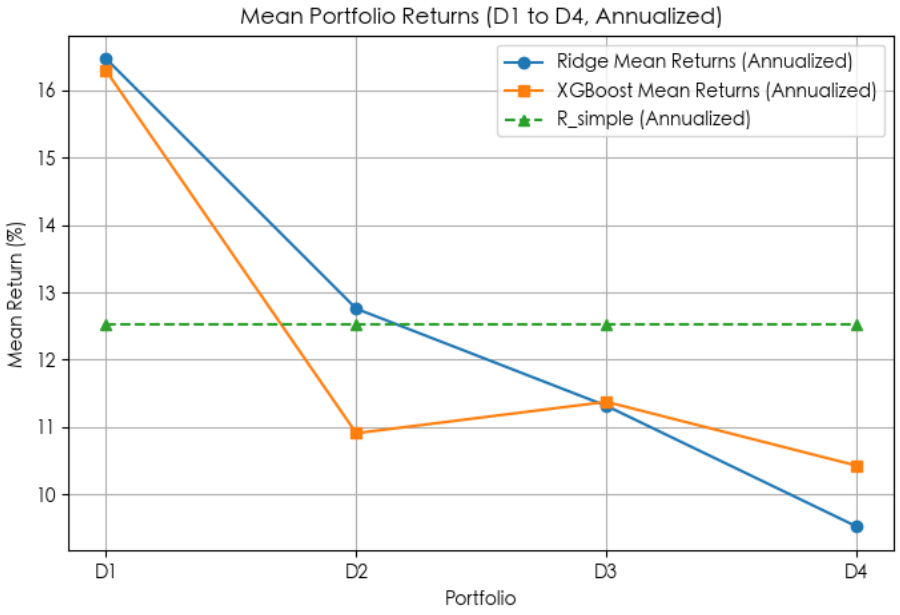
新參數的	D1	D2	D3	D4
Mean	16.657393	12.739818	10.989164	9.666944
Volatility	71.005237	52.386229	40.813131	41.453944
Mean-to-Volatility Ratio	0.234594	0.243190	0.269256	0.233197

XGBoost：

新參數的	D1	D2	D3	D4
Mean	13.582893	12.279233	10.562634	13.202920
Volatility	61.889434	45.556786	46.969405	51.675500
Mean-to-Volatility Ratio	0.219470	0.269537	0.224883	0.255497

就圖表來說，Ridge 應該是訓練的不錯的，至少我預測出來的平均報酬有滿足 D1>D2>D3>D4，但 XGBoost 就沒有，而且 D1 跟 D4 的報酬基本上是差不多的，所以這也導致在做 RLS 投資組合時，算出來的結果特別糟糕，所以 XGBoost 的效果會很差，我覺得應該是跟我的參數選取有關，這次放寬參數的選取，但實際做出來的

結果是比之前 Part1 做出來的還要糟糕的（下面提供 Part1 的圖表），就是說可以很明顯的發現 Part1 在做的時候，D1 跟 D4 並沒有很相同的結果，有明顯的差距，雖然說最後結果沒有滿足 D1>D2>D3>D4 的遞減關係，但至少有比這次新參數的結果來的好，所以我會覺得可能是參數的選擇沒有選好導致 XGBoost 的結果比上次 Part1 來得更糟糕。但對於 Ridge 的部分，以圖表來說新舊結果是差不多的，我觀察起來是 D1 跟 D4 的平均報酬差距沒有到非常大，也就是說如果我們將群組分的在更細一點，可能取到八組之類再做 RLS 那算出來的結果應該是有機會超越 benchmark 的，因為就個別 D1-D4 的 Mean-to-Volatility Ratio 是沒有輸 benchmark 太多的，所以我覺得仔細來看，可能 Ridge 訓練出來的結果並沒有不理想，可能只是在更細分一下組別就能打敗 benchmark 了。



（上圖）Part1 算出來的 D1-D4 圖

XGBoost：

Part1	D1	D2	D3	D4
Mean	16.291336	10.907912	11.374097	10.426837
Volatility	62.353481	46.820809	43.557607	52.091871
Mean-to-Volatility Ratio	0.261274	0.232971	0.261128	0.200162

(b) Model interpretation: What are the relationships between the best predictor and stock returns during the best and worst portfolio performance? (This is Question 4 in Part I)

我先附上這次算出來的表格跟之前 Part1 做出來的表格，我們再來一步一步的做後續討論，看看問題有哪些不一樣。

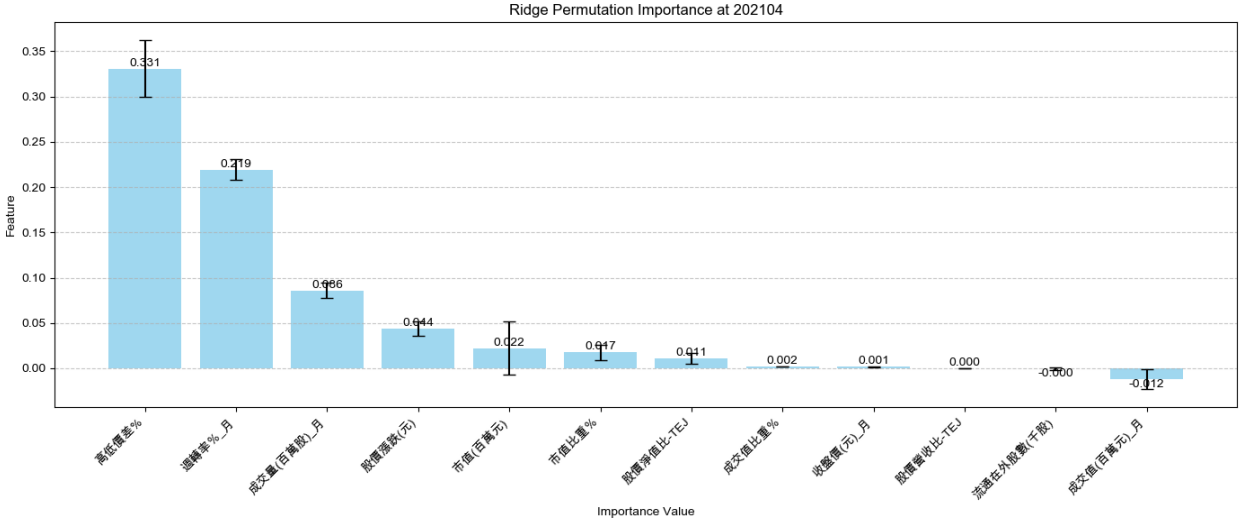
新參數的	t^{max}	Hyperparameters	t^{min}	Hyperparameters
Ridge	2021-04-01	50.0	2021-07-01	50.0
XGBoost	2021-06-01	Max depth: 3, learning rate: 0.01, n estimators: 400	2020-11-01	Max depth: 3, learning rate: 0.01, n estimators: 300

Part1 的	t^{max}	Hyperparameters	t^{min}	Hyperparameters
Ridge	2021-04-01	10.0	2021-05-01	10.0
XGBoost	2021-04-01	Max depth: 6, learning rate: 0.1, n estimators: 300	2020-04-01	Max depth: 7, learning rate: 0.1, n estimators: 300

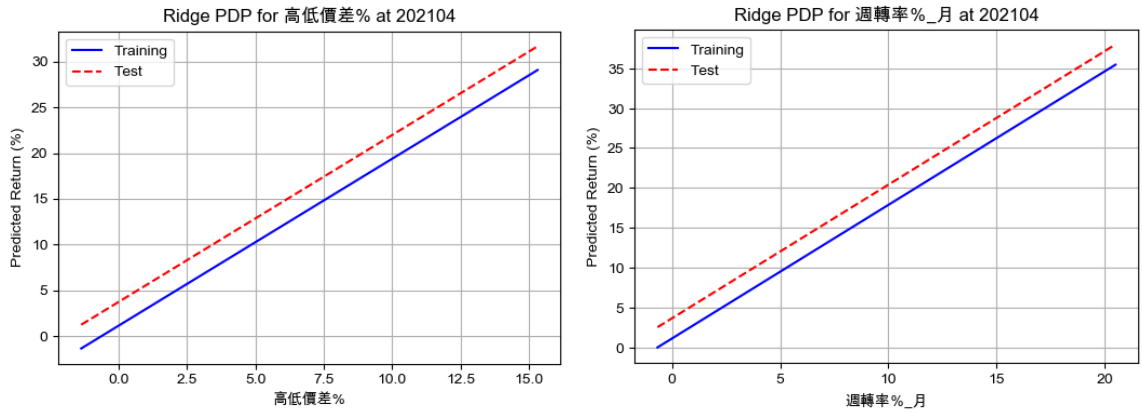
這樣的結果應該算是我不想看到的，當初對 Ridge 的 alpha 提高參數的選擇就是希望不要碰到邊界，對於 XGBoost，也有同樣的煩惱，希望開放參數的選擇來避免沒有達到最佳訓練的問題，結果這次的 alpha 選擇還是 50，是我所提供的參數裡最大值，所以依舊沒有解決這樣的困擾，而 XGBoost，反而有部分參數選擇了可供選擇裡的最小值，所以也有可能會有類似的問題依舊存在。但算出來的最佳與最差結果時間有不一樣，所以也參數的答案會不同，也有可能是此原因造成的。

那我們接著討論到底有哪些因子在影響我的報酬，所以我去分別計算他們的 Important Value，想觀察哪些因子是相對重要的。

我們先來看看 Ridge 的模型：

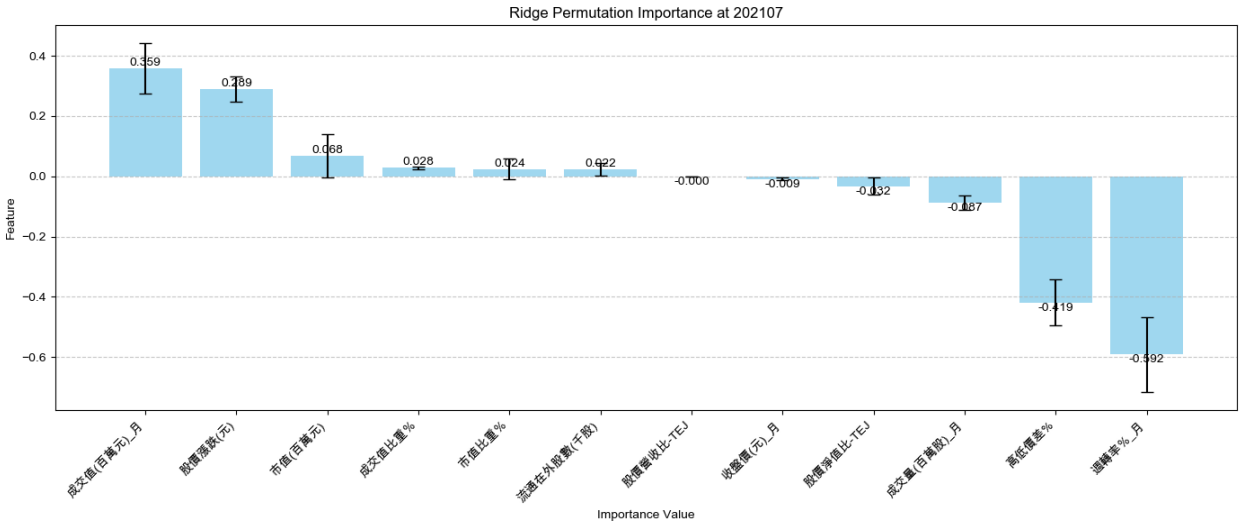


（上圖）是 Ridge 在t^{max}時的 Important Value

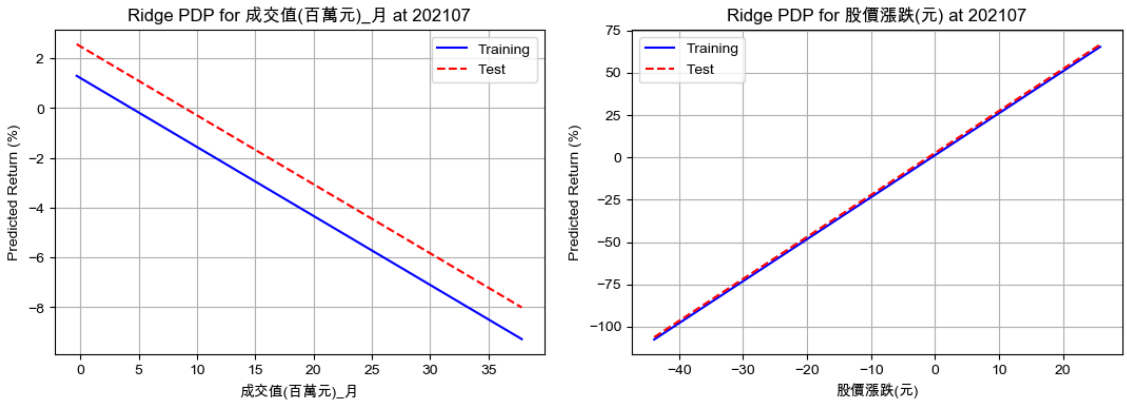


上圖為 Ridge 表現最佳時前兩個最佳因子的 PDP

在t^{max}這個時間點來說，影響成分最重的是，「高低價差%」跟「週轉率%_月」，這結果跟之前 Part1 做出來的結果是一樣的，我自己是覺得這兩個因子去解釋模型還算可以理解，而且觀察 PDP 的圖表也發現他有很強的正相關，我們有較高的「高低價差%」表示說這股票某種程度上有較高的潛力，然後可能大家買賣的交易量也是相對較大的，所以才會有明顯的「高低價差%」，那因為「週轉率%_月」高，表示說股票有好的流動性，這也再次表示這家公司有好的交易量，所以會預期未來可能會有好的投資報酬率。



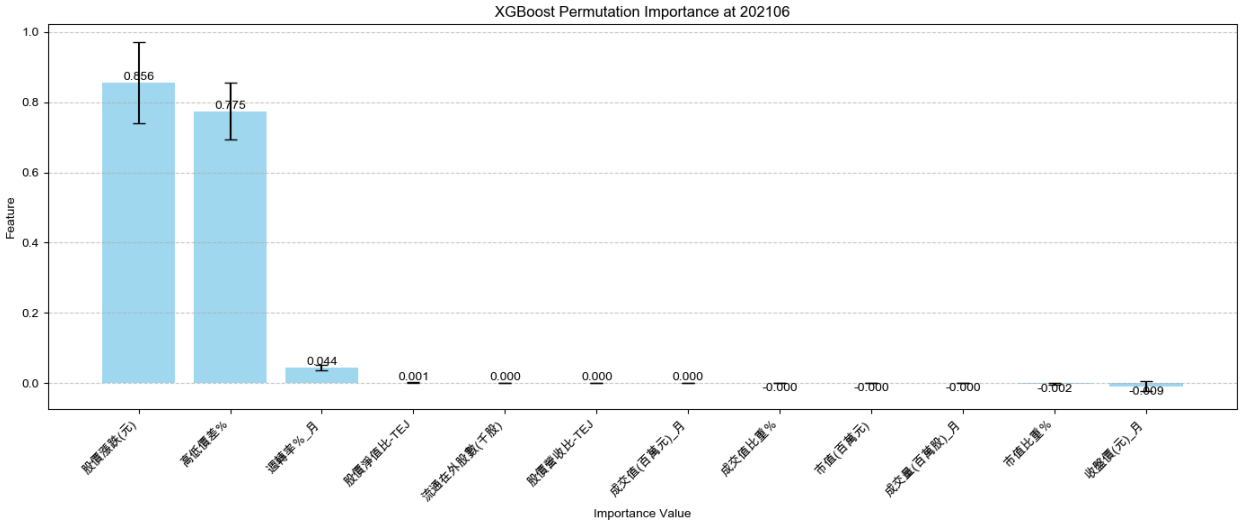
（上圖）是 Ridge 在t^{min}時的 Important Value



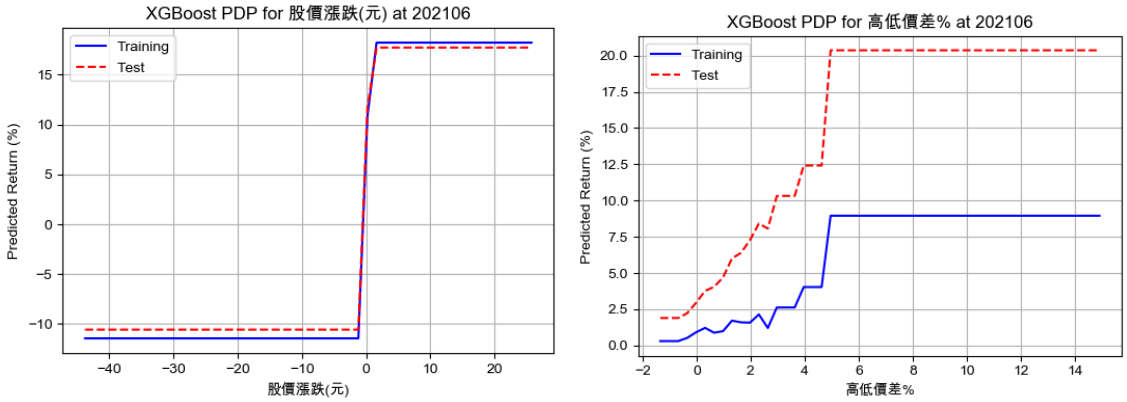
上圖為 Ridge 表現最差時前兩個最佳因子的 PDP

那現在看到t^{min}這個時間點，我們原本就比較重要的兩個指標，「高低價差%」跟「週轉率%__月」在這時間點算出來的 Important Value 卻是負的，表示說明明對於線性模型 Ridge 最重要的因子現在在這個時間像是在搞事，沒有好的預測能力，那相對的，這時候預測出來的結果當然不會太好，這也是可以理解的。然後觀察前兩個重要因子的 PDP，可以發現「成交值（百萬元）_月」明明最重要的影響因子，但他卻是有很強的負相關性，所以這也表示很有可能是因為這個原因導致我的預測結果非常差。

接下來，我們來看看 XGBoost 的模型：

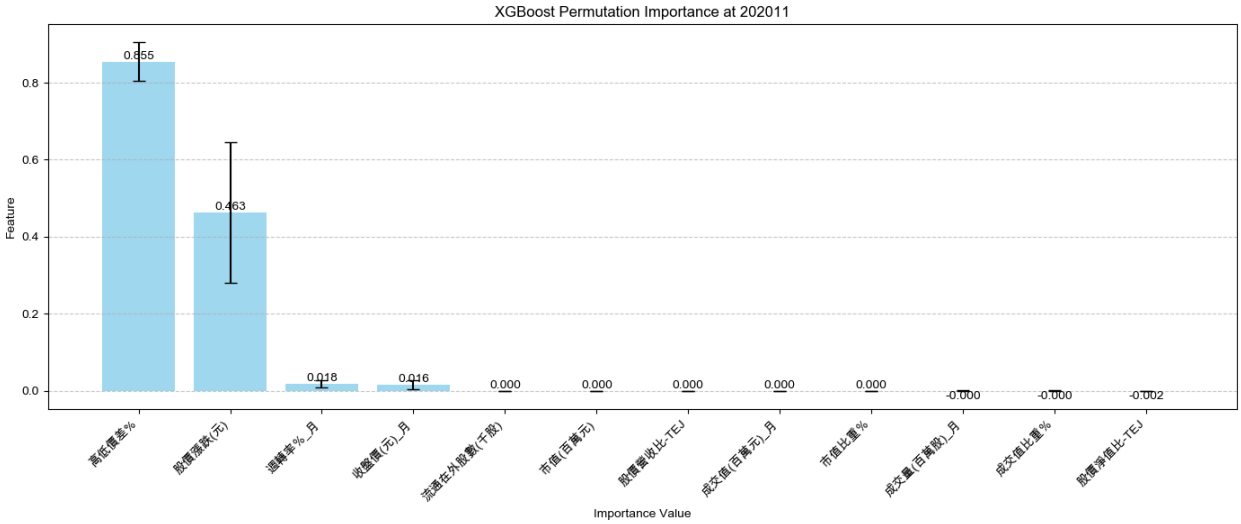


（上圖）是 XGBoost 在t^{max}時的 Important Value

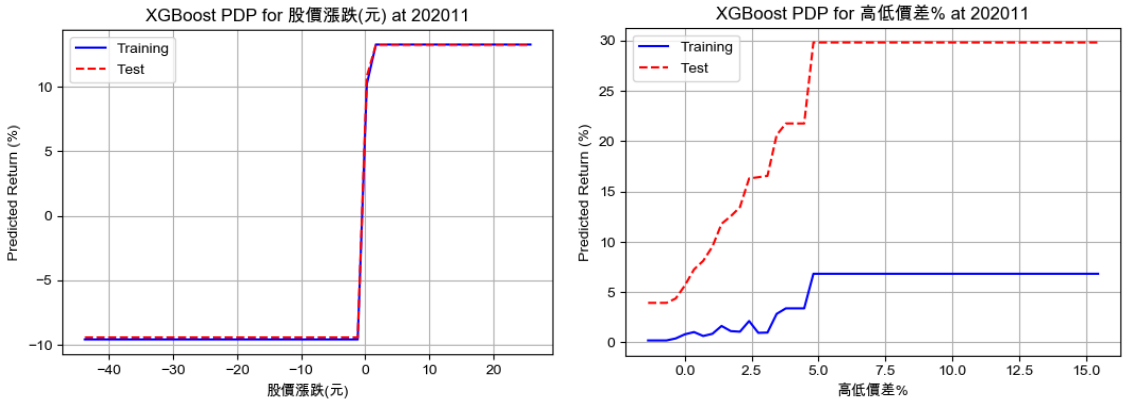


上圖為 XGBoost 表現最佳時前兩個最佳因子的 PDP

在t^{max}這個時間點來說，影響成分最重的是，「高低價差%」跟「股價漲跌（元）」，這結果跟之前 Part1 做出來的結果是一樣的，而且觀察 PDP 的圖表也發現他有很強的正相關，我自己是覺得這兩個因子去解釋模型也是可以理解的，「高低價差%」的部分就如前面所講的一樣，至於「股價漲跌（元）」就是很技術指標的分析，通常我們會預期前一天大跌的股票隔天也會是下跌的，反之，前一天大漲的股票，我們也會預期隔天股價會上漲，所以就結果來看，這兩個因子去做股票預測也算是解釋得通。然後其他因子在這個時間點下，對於模型的影響力近乎是 0，這也算是一個蠻特別的情形，跟線性模型的結果有點不一樣。關於 PDP 我想再補充一下，「股價漲跌（元）」的 PDP 圖非常合理，我只要有一些些負值，那就會預測有負報酬，有一些些正值，那就會預測有正報酬，這跟前面提到的理論很符合結果。



（上圖）是 XGBoost 在t^{min}時的 Important Value

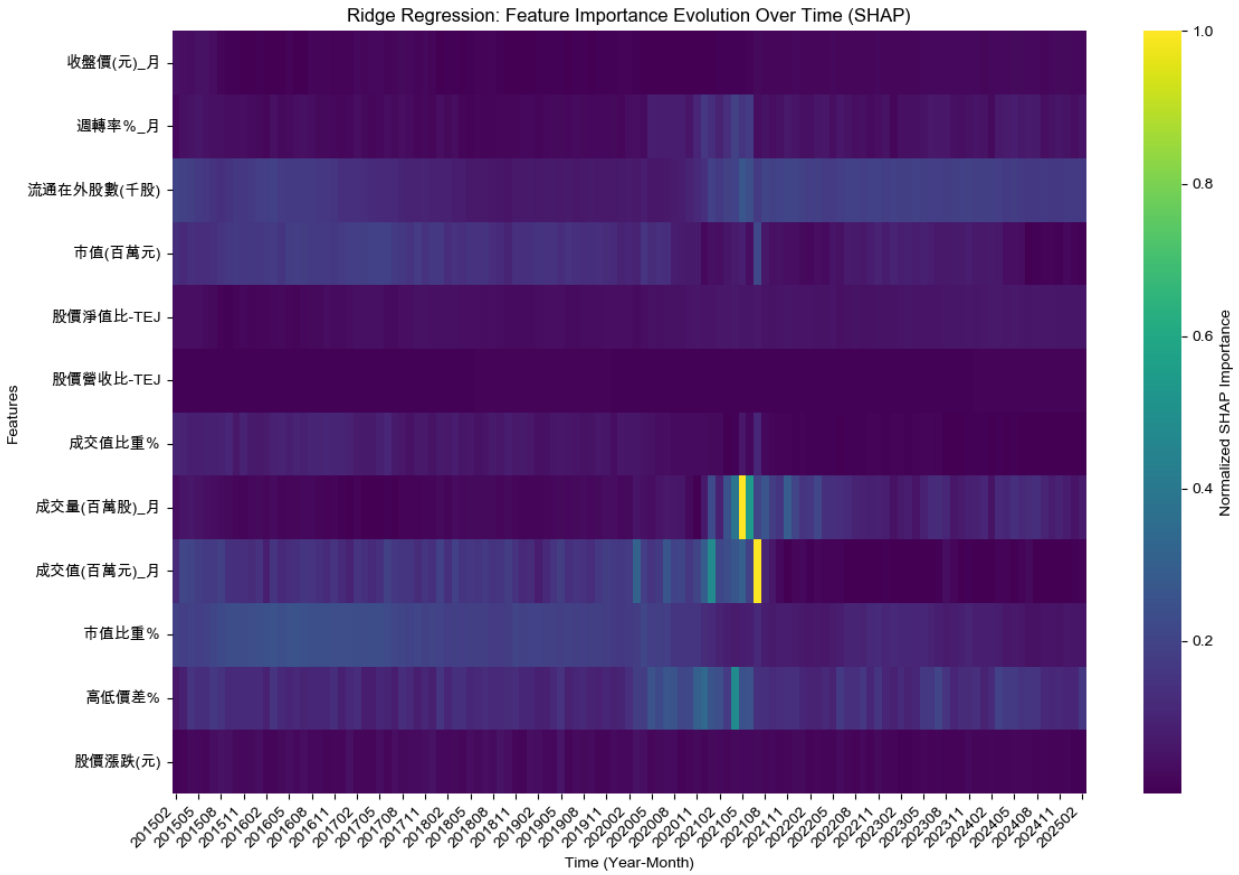


上圖為 XGBoost 表現最差時前兩個最佳因子的 PDP

在t^{min}這個時間點來說，影響成分最重的是，「高低價差%」跟「股價漲跌（元）」，這結果跟之前 Part1 做出來的結果是一樣的，那可能就是因為該時間的時候這兩個因子的數值都不好，所以導致最後計算出來的預測報酬很差。然後其他因子在這個時間點下，對於模型的影響力近乎是 0，這也算是一個蠻特別的情形，跟線性模型的結果有點不一樣。這時間點的 PDP 圖其實跟最佳時間點的圖差不多，所以我猜就可能真的是這個時間點「高低價差%」跟「股價漲跌（元）」的樹樹真的不佳，所以才會跟著預測出不好的報酬結果。

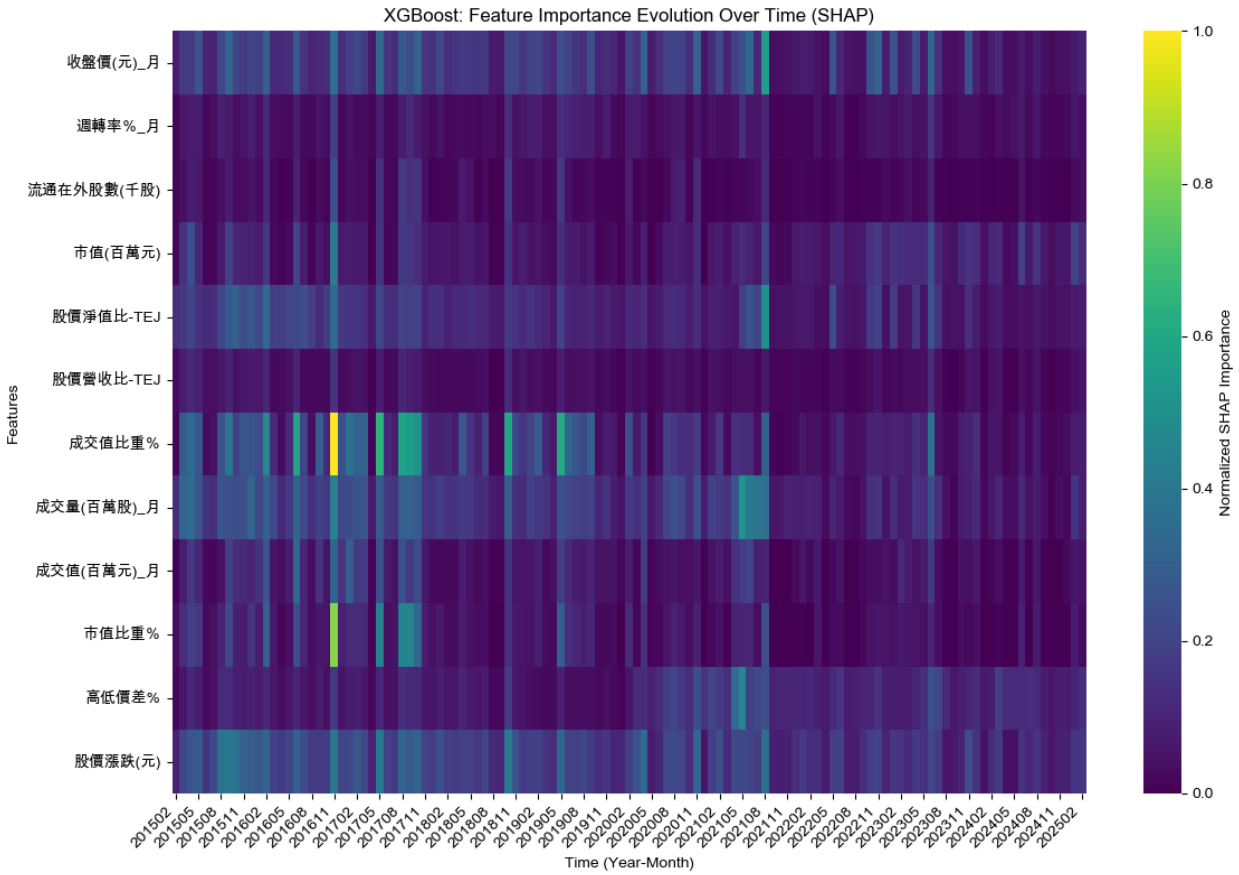
(c) The evolution of feature importance across time: Plot a heatmap (see the next page for an illustrative example) that shows the feature importance (SHAP or permutation-based) across time (This is not asked in Part I). Please discuss.

我這裡是計算從 2015 年 02 月開始畫然後到 2025 年 02 月的熱圖，然後都是對每個因子的 Shap 值去做生成的，深紫色為最低重要性數字是 0，亮黃色為最高重要性數字是 1。



（上圖）是 Ridge 訓練出來的熱圖

由上圖可知”成交值(百萬元)_月”、”高低價差%”、”週轉率%_月”，這三個特徵在多數的時間點都呈現較高的 SHAP 值（偏綠～黃），表示 Ridge 模型在預測股票報酬時，這些特徵有穩定的影響力，所以報酬預測的好壞可能會有很大的關係跟這三個特徵表現的好壞有關。然後”股價營收比-TEJ”、”股價淨值比-TEJ”、”股價漲跌（元）”、”收盤價（元）_月”，這四個特徵在幾乎所有時間點都偏深紫色，表示 Ridge 模型在預測股票報酬時，這些特徵幾乎都沒有影響力，所以之後可以考慮從特徵集裡移除這四個參數。然後，”成交量（百萬股）_月”、”成交值（百萬元）_月”，這兩個因子有部分時期明顯表現不錯，部分時期明顯表現較差，這表示這些因子對預測有影響，但不像”成交值(百萬元)_月”、”高低價差%”、”週轉率%_月”這三個因子是明顯的每個時間點都有作用。



（上圖）是 XGBoost 訓練出來的熱圖

由上圖可知”收盤價（元）_月”、”股價淨值比-TEJ”、”成交值比重%”、”成交量（百萬股）_月”、”股價漲跌（元）”，這五個特徵在多數的時間點都呈現較高的 SHAP 值，表示 XGBoost 模型在預測股票報酬時，這些特徵有穩定的影響力，所以報酬預測的好壞可能會有很大的關係跟這五個特徵表現的好壞有關。然後”週轉率%_月”、”流通在外股數（千股）”、”股價營收比-TEJ”，這三個特徵在大部分的時間點都比較偏深紫色，表示 XGBoost 模型在預測股票報酬時，這些特徵幾乎都沒有影響力，所以之後可以考慮從特徵集裡移除這三個參數。

雖然說熱圖可以以顏色表現去區分特徵的影響力，所以 XGBoost 的熱圖並沒有 Ridge 的熱圖那麼一致明顯的都是長期在特定幾個特徵上時，這是我們所期望的結果，而這也表示 XGBoost 有去利用不同的因子特性去預測下一期的報酬。這也明顯是非線性模型的特性，不會特定依靠某些特徵去做預測，所以這表示非線性的模型可以更佳彈性的去適應市場模型，只是很可惜，在這次的訓練結果裡，我自己是覺得 Ridge 的表現是比 XGBoost 的還來的更好，因為在 D1-D4 的分類效果下，Ridge 的模型明顯是有達到較佳的分類預測結果。

4. Conclusion

我自己覺得這份報告寫完，我不能說線性或者非線性就一定有比另一者還要好，因為就 RLS 的結果來看，確實是線性模型佔優勢。可就熱圖的分析結果來看，我自己覺得非線性模型的優勢有展現出來，只是可能因為參數選擇較多，所以並沒有選擇到最為合適的參數去預測，進而導致預測結果不佳。而已 Important Value 跟 PDP 圖表去分析的話，我自己覺得個別都解釋得通，但我自己是偏好更喜歡 XGBoost 的結果的，因為可能我被「股價漲跌（元）」的 PDP 圖影響，我覺得這張圖所表現的非常符合我的經濟直覺，不是說其他圖沒有，只是我會覺得哪怕是正相關，應該也會有個上限，不可能說我特徵數值越大，預測報酬就越大趨近於無限，總會有個適當的上限，就結果應該會想像像是 Sigmoid 的圖，所以我自己會覺得喜歡 XGBoost 的結果，但他需要給予其他參數選擇，讓他可以達到好的訓練結果。