

元大證券計量交易部實習期末報告

見習單位：元大證券 計量交易部

見習主管：余光麒 資深副總經理

見習督導：曾盟雅 專業經理、郭獻聰 學長

實習生：陳冠維 國立清華大學計財系

專案主題：
運用深度學習模型預測個股之報酬

研究大綱

一、預測個股之報酬

1. LSTM 模型
2. 預測個股報酬
3. 回測結果
4. 小結

二、選取重要因子

1. 深度學習選取因子方法
2. 選取因子結果、發現
3. 回測結果
4. 小結

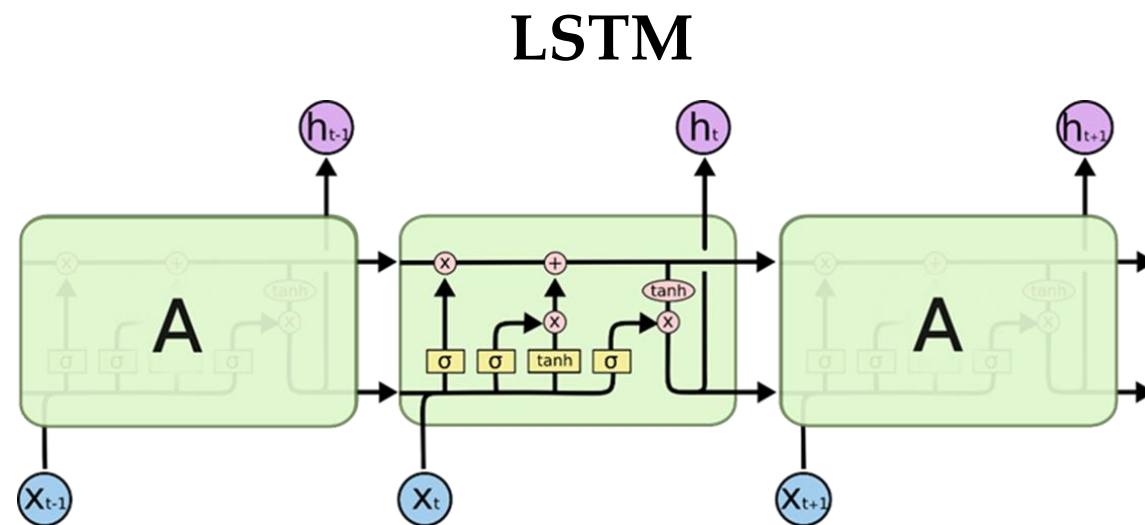
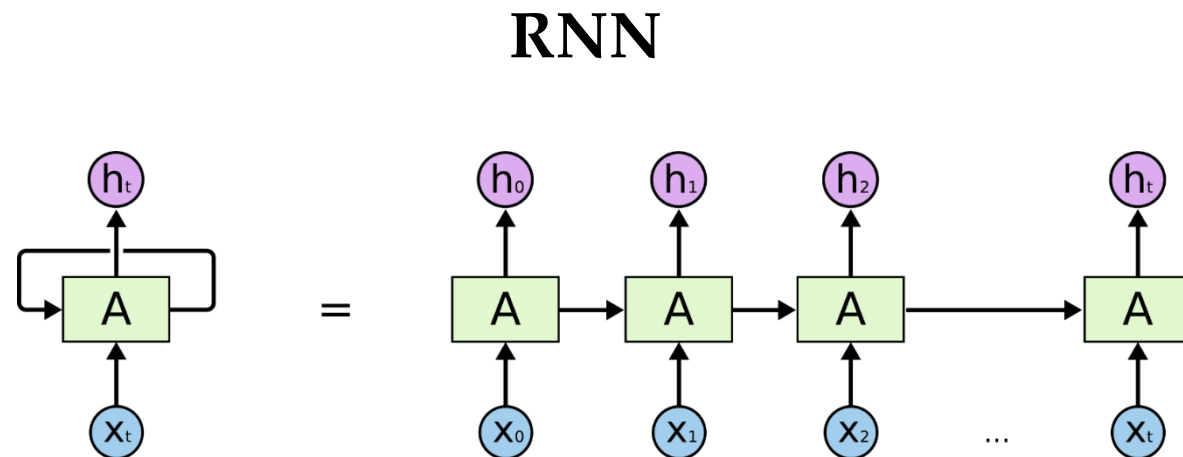
三、結合lasso模型

1. 尋找重要因子
2. 結合LSTM
3. 回測結果
4. 小結

一、預測個股之報酬

傳統RNN模型（遞迴神經網路）

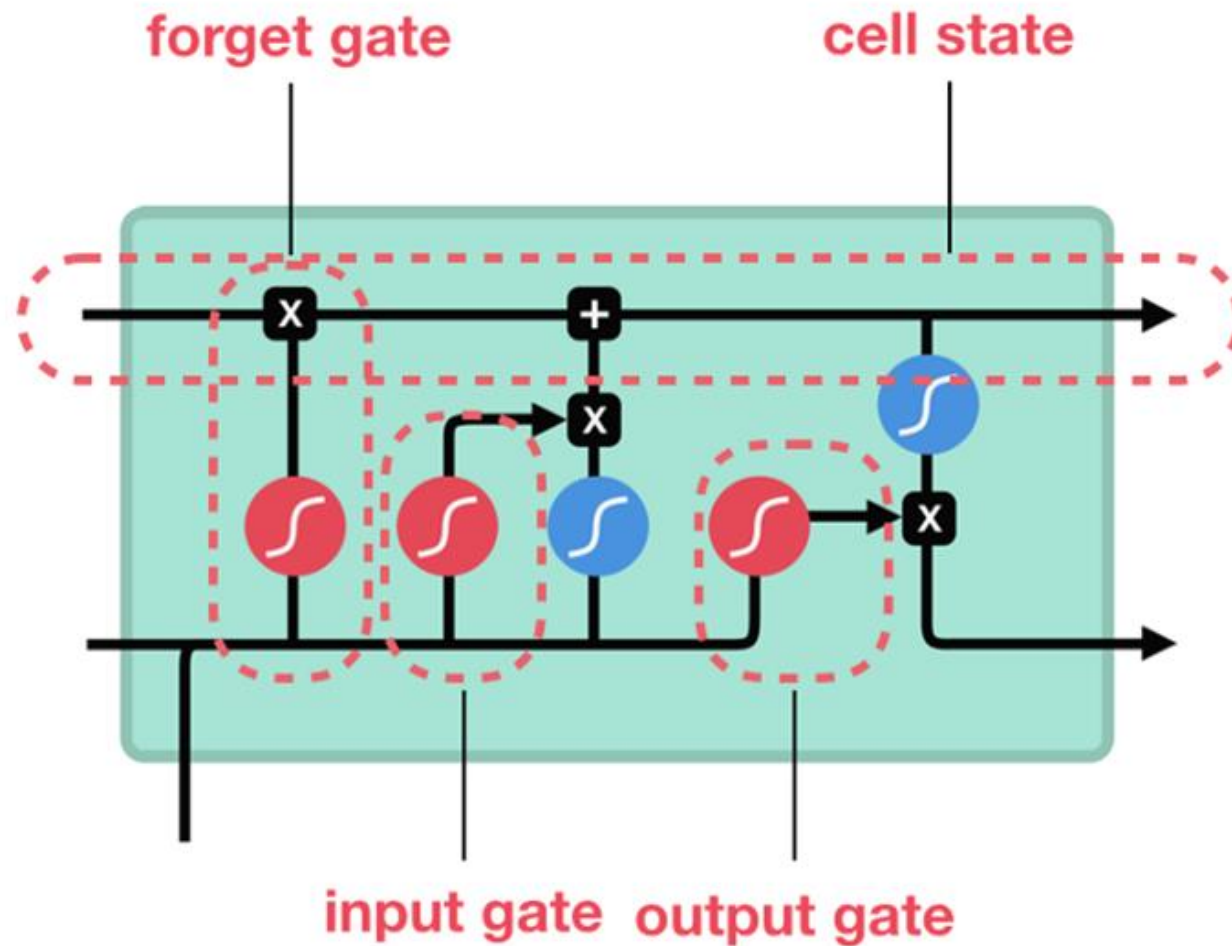
- **RNN 處理資料的特性:** RNN 不會一次將整個序列 X 讀入，而是像迴圈一般一次讀入一個小 x ，並在每個時間點更新細胞 A ，並輸出結果小 h 。
- 優點: **處理序列資料**，例如: **文本**資料
- **RNN 的記憶狀態:** RNN 在每次讀入任何新的數據之前，細胞 A 中的記憶狀態都會被**初始化為0**，這導致 RNN 沒辦法很好地「記住」前面處理過的序列元素，造成 RNN 在處理後來的元素時，就已經把前面重要的資訊給忘記了。
- 缺點: RNN**無法記住前面處理過的元素**，LSTM可以！



LSTM模型

- **Forget Gate:** 決定細胞是否要遺忘目前的記憶狀態
- **Input Gate:** 決定目前輸入有沒有重要到值得處理
- **Output Gate:** 決定更新後的記憶狀態有多少要輸出

➡ 優點: 處理時間序列資料



傳統 RNN VS LSTM

傳統RNN

- 語言閱讀的思路，接近於人。
- 在實作上，傳統的 RNN 很難捕捉到長期的記憶，數學上所產生的梯度消失的問題造成長時間的記憶會被短時間的記憶所隱藏。
- 長時間的記憶表現不好，常只考慮最近的輸入。

LSTM

- LSTM 透過設計較佳的激勵函數，在神經單元(細胞A)中加入遺忘、輸入(更新)、輸出閥門，去決定哪些細胞狀態應該被遺忘、哪些新的狀態應該被加入、根據當前的狀態和現在的輸入，輸出應該要是什麼。
- 增加長時間的記憶表現，不只考慮最近的輸入，可以將任意時間點的狀態拿來使用。
- 運用於個股價量的時間序列資料。

預測個股報酬：參考論文

- 因子參考論文

Modeling high-frequency limit order book dynamics with support vector machines

- 模型參考論文

A LSTM-based method for stock returns prediction: A case study of China stock market

Study of Stock Return Predictions Using Recurrent Neural Networks with LSTM

- 建模參考

<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

<https://www.kaggle.com/amarpreetsingh/stock-prediction-lstm-using-keras>

https://github.com/Danjtchen/LSTM_stock_example/blob/master/LSTM_example.ipynb

預測個股報酬: 因子

Time-insensitive Set

- Bid-Ask spread
- Mid Price
- Price Difference
- Mean Price & Mean Volume
- Accumulative Bid-Ask spread
- Accumulative Bid-Ask Quantity spread

Time-sensitive Set

- 1-tick 價格差分
- 5-ticks 價格差分
- 10-ticks 價格差分
- 1-tick 量取差分
- 5-ticks 量取差分
- 10-ticks 量取差分

預測個股報酬：資料型態

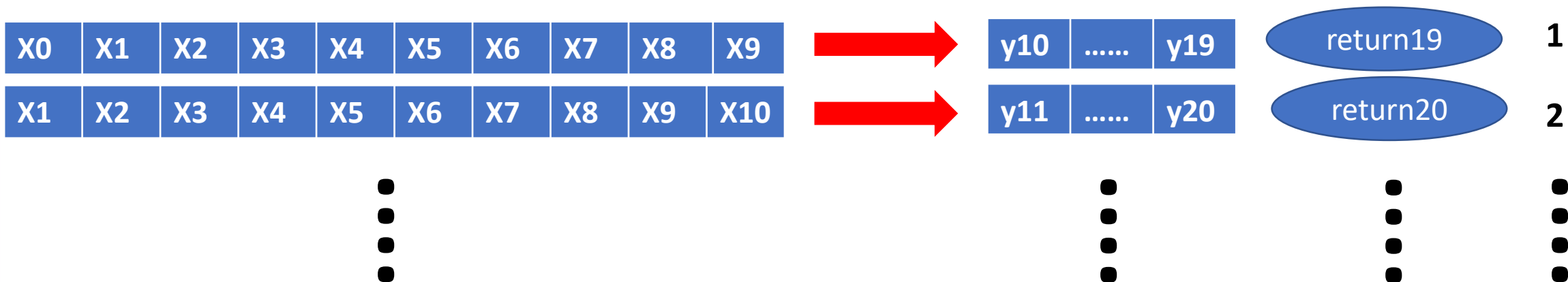
Target: 玉晶光 (3406)

Train: 3/25 ~ 4/23 (19 days) *Test:* 4/24 ~ 5/5 (7 days)

Response (Y): 後10個ticks 的報酬 (return_{10t})

Covariates (X): 前述的全部因子

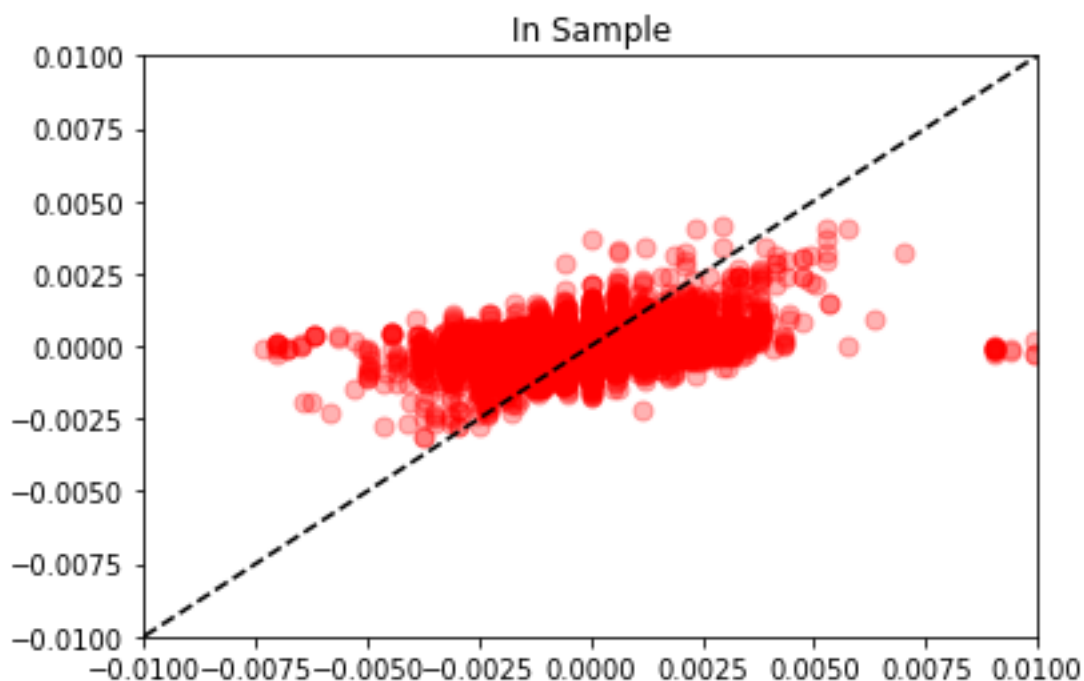
Time Periods: 運用前10個ticks的X，預測後10個ticks 的報酬



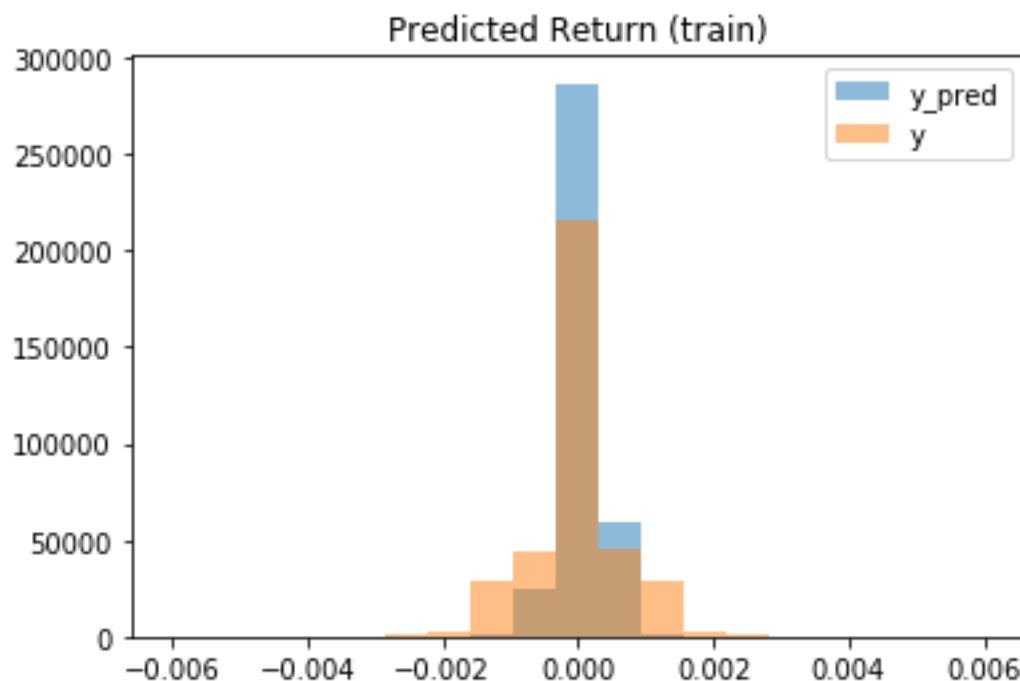
預測個股報酬: LSTM預測下10個ticks的股價報酬 (3406 train)

Train R-squared: 0.12

真實的報酬與預測的報酬的散布圖



真實的報酬與預測的報酬的直方圖



預測個股報酬：策略邏輯

Strategy Concept:

1. 以預測出的未來10個ticks後的報酬為基準
2. 依據此報酬判斷做空、做多 ($\pm 0.1\% \sim \pm 0.3\%$)
3. 手上部位保持在 $[1, 0, -1]$ ，當日收盤前須結掉手上的單

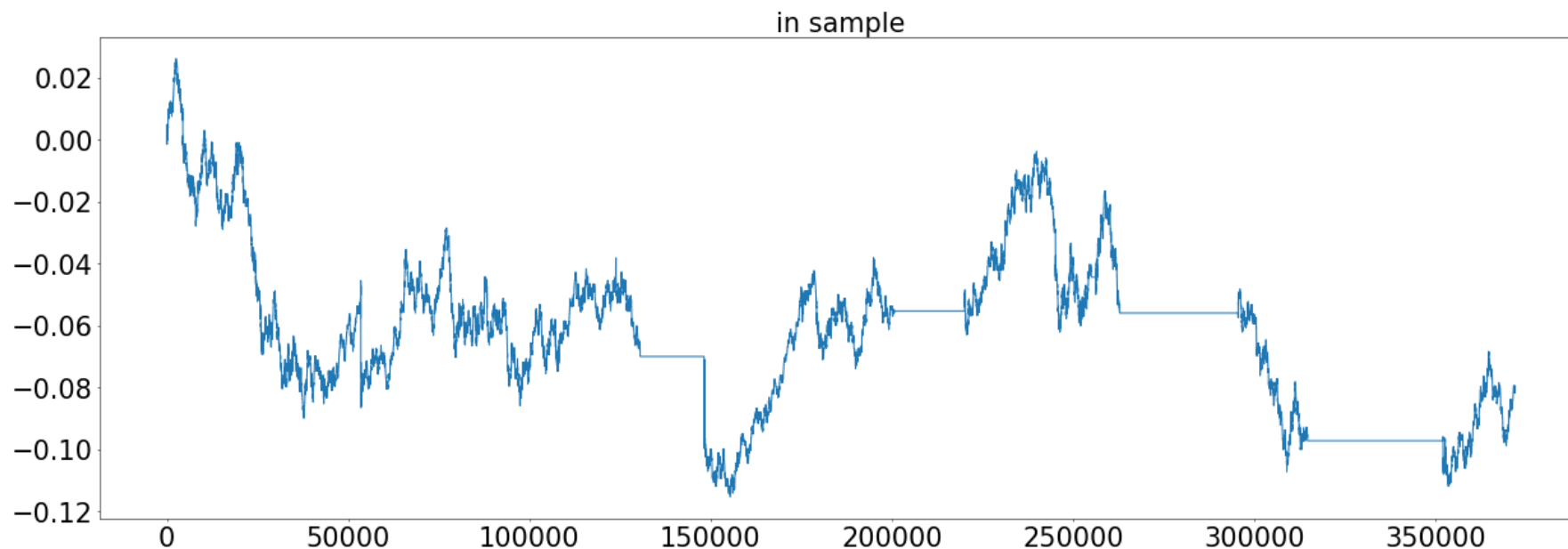
預測個股報酬: 回測結果 (3406 in sample)

Parameters

1. 預測未來 10個tick 漲 0.15% 做多
2. 預測未來 10個tick 跌 0.15% 做空

Performance of train set

- Return: -8.3%
- Vol: 0.024
- Sharp Ratio: -3.7
- Transactions: 51
- Odds Ratio: 0.39
- Max Drawdown: 0.103



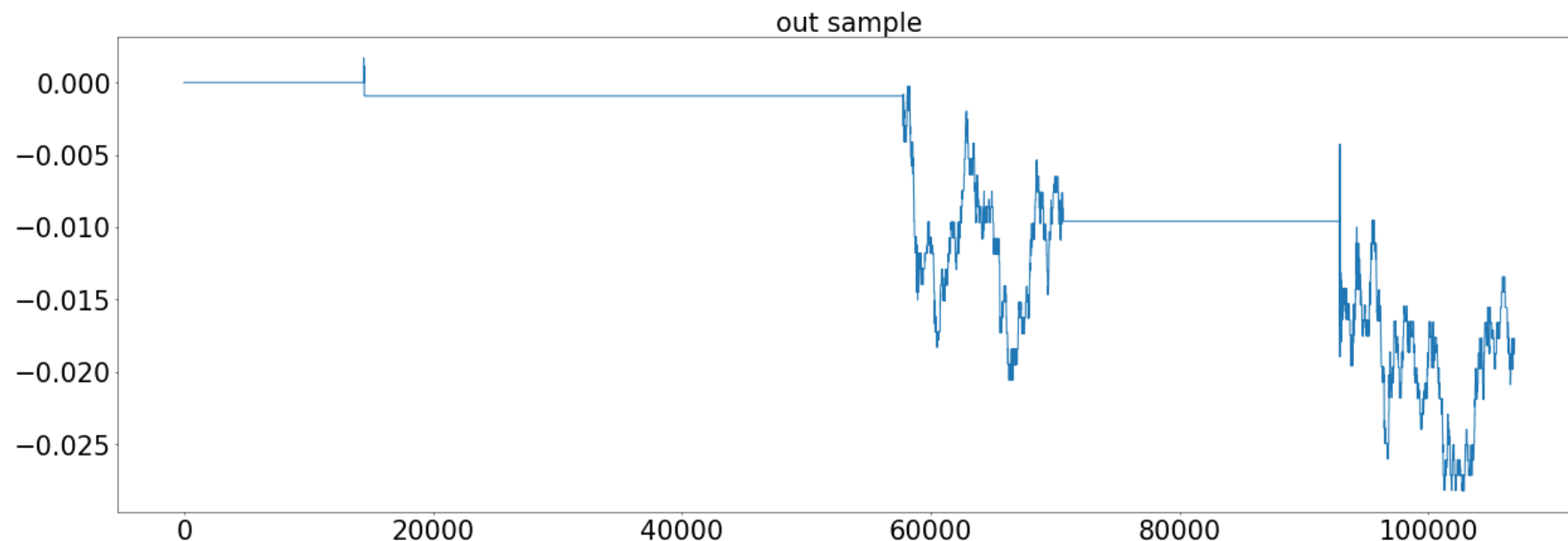
預測個股報酬: 回測結果 (3406 out sample)

Parameters

1. 預測未來 10個tick 漲 0.15% 做多
2. 預測未來 10個tick 跌 0.15% 做空

Performance of test set

- Return: -2%
- Vol: 0.007
- Sharp Ratio: -3.6
- Transactions: 4
- Odds Ratio: 0
- Max Drawdown: 0.022



預測個股報酬: 小結

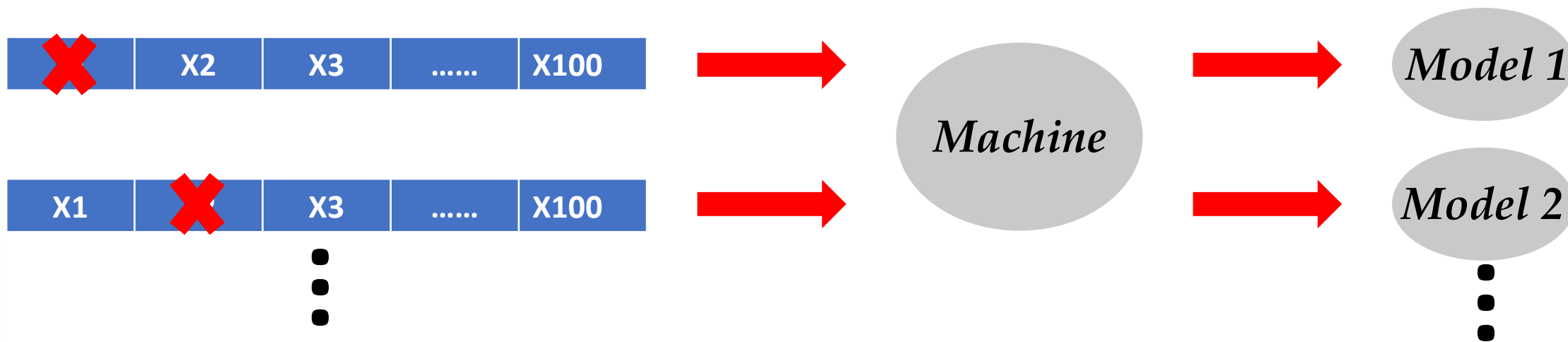
1. 模型預測報酬的準確度不足，常常低估真實的股價報酬，造成真實值與預測值有一段落差。
2. 根據統計學的理论，資料量增加可以抵銷掉Over-fitting 的結果。畢竟台股在3月底才開始逐筆搓合，如果未來資料量能增加，相信模型能夠更好的學到個股報酬的趨勢，在樣本內跟樣本外都有好的展現。
3. 在做了幾個月的研究後，發現模型再怎麼調整，實際上差不了太多。比較重要的是我們丟了甚麼因子到模型裡。因此如何去挑選有用的因子我認為是很重要的。因此在實習後半部我開始針對重要的因子去做開發。

二、選取重要因子

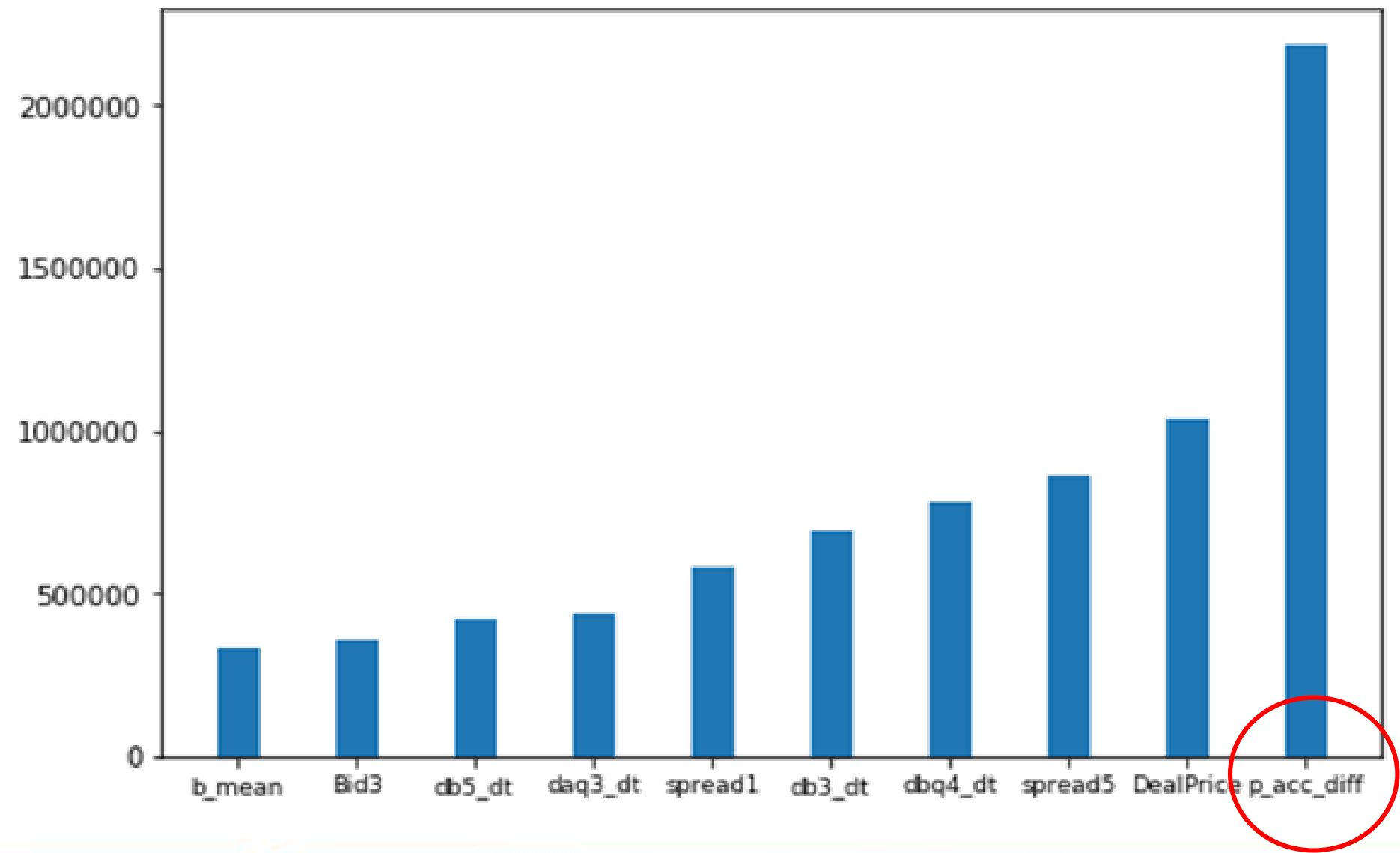
選取重要因子: 深度學習選取因子方法

Drop One Out

1. 先用全部的因子訓練出一個完整的模型，算出 *test MSE*
2. 每一次將一個因子丟掉，每一次都訓練出一個模型，算出 *test MSE*
3. 比較每一個模型的 *test MSE*，*test MSE* 增加最多的那個模型所丟掉的因子選為最重要的因子

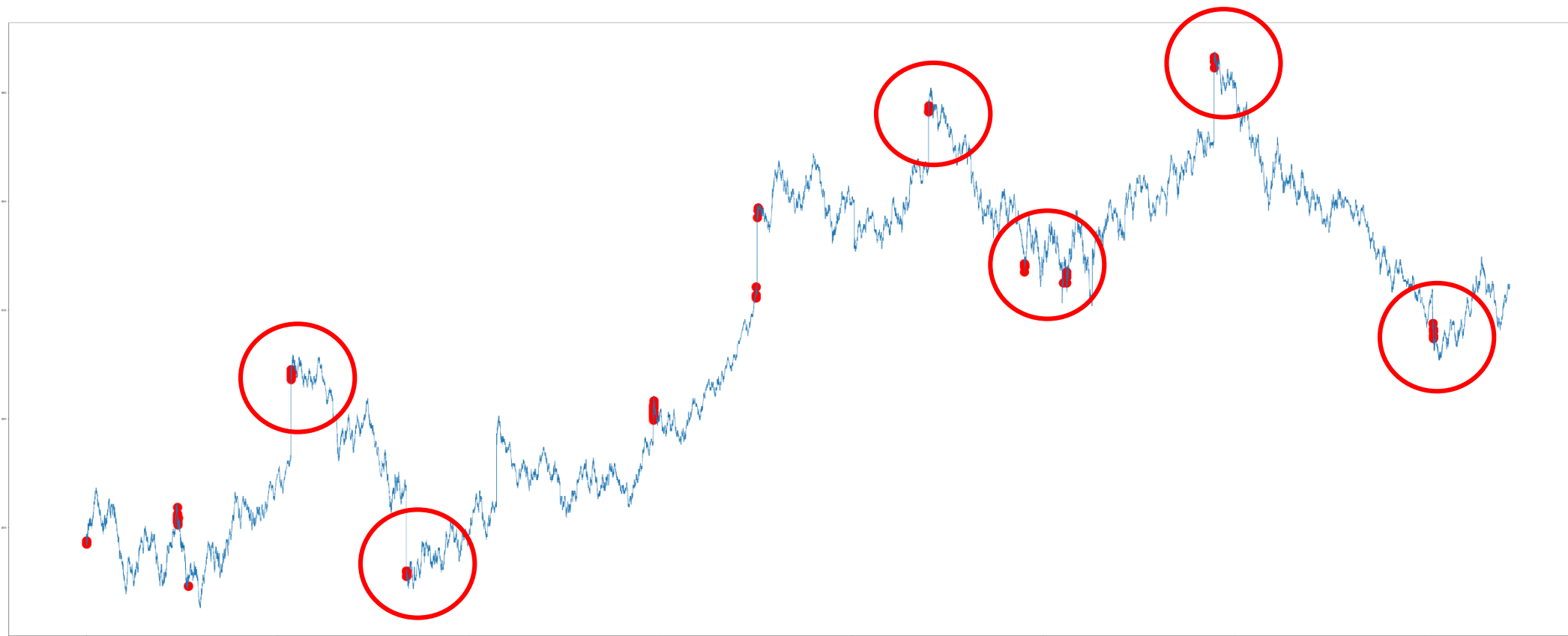


選取重要因子: 選取因子結果



選取重要因子: 選取因子發現

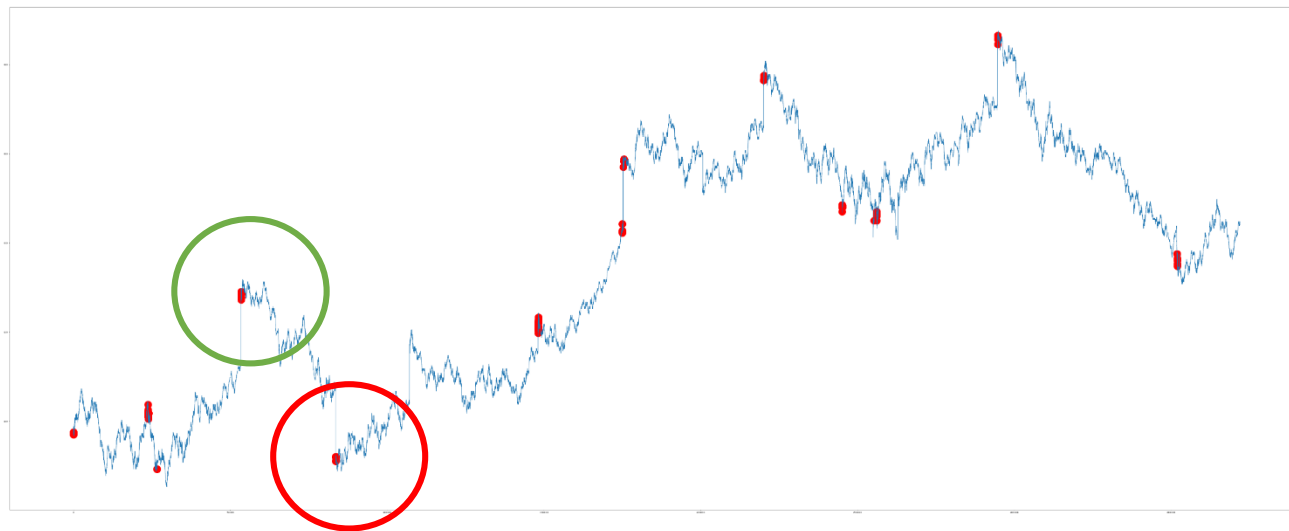
- *Variable Importance*: **p_acc_diff** (5檔 Bid Ask spread 之和)
- 挑選出5檔Bid Ask spread 之和大於25的時間點，畫在訓練集的中價走勢圖上



選取重要因子: 策略邏輯、回測結果 (in sample)

Strategy Concept:

1. 將 $p_acc_diff > 25$ 當作一個濾網
2. 加上50個ticks的MA 判斷中價走勢
3. $p_acc_diff > 25$ & 中價 < 前50個ticks的MA時，做多
4. $p_acc_diff > 25$ & 中價 > 前50個ticks的MA時，做空



Performance

- Return: 16%
- Vol: 0.048
- Sharp Ratio: 3.19
- Transactions: 34
- Odds Ratio: 0.29
- Max Drawdown: 0.042

選取重要因子: 小結

1. 這個因子反應了價格走勢反轉的資訊，確實可以當作一個濾網來做交易策略，在未來的研究上還可以用上。
2. 單純以這個濾網做的策略雖有獲利，但勝率不高，因為常常會捕捉到一些雜訊，可能還需要加上其他的因子去做過濾。
3. 雖然在訓練集有獲利，但在測試集裡幾乎沒有訊號產生。有可能是因為這個濾網只在某段時間有用，未來還要再觀察這檔股價的走勢才能定論。
4. 因為我用drop one out的方式選取重要因子，每次訓練出來的模型不同，因此選出來的重要因子可能不一定是最好的，再加上深度學習訓練耗時，也耗GPU，因此之後我會結和學長較為簡單的回歸模型去做因子的挑選。

三、結合LASSO模型

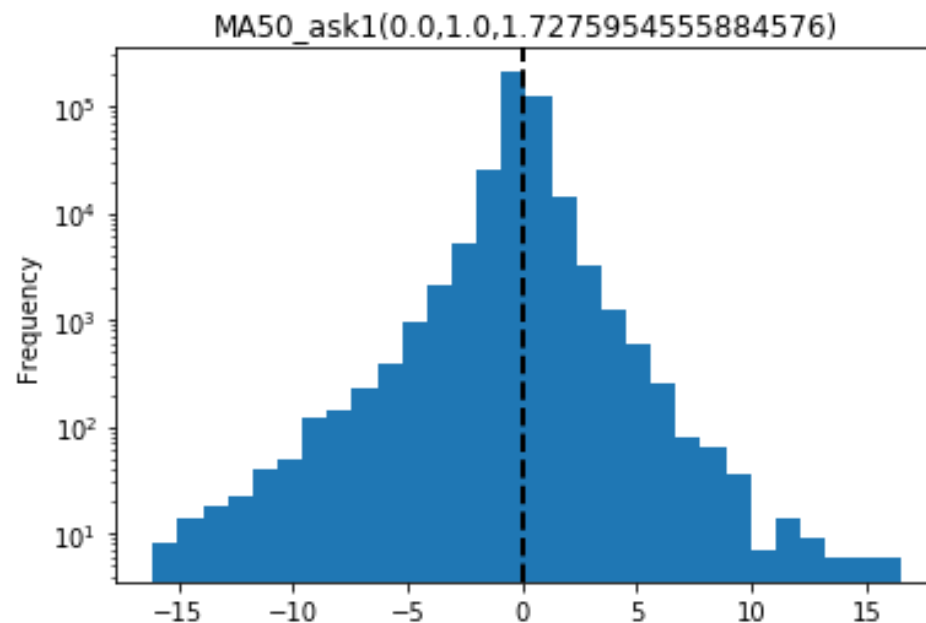
結合lasso模型: 因子

5檔買賣邊的力道 (Value)

- Ask Price x Ask Quantity
- Bid Price x Bid Quantity

Moving Average – Value

- Ask Value 50 ticks 的 MA – Ask Value
- Bid Value 50 ticks 的 MA – Bid Value
- Ask Value 100 ticks 的 MA – Ask Value
- Bid Value 100 ticks 的 MA – Bid Value



因子與報酬的主觀解釋

- MA_ask: 當Ask Value 過去50個ticks 的平均大於現在的Ask Value 時，代表現在市場處於多頭，與報酬呈現正相關。
- MA_bid: 當Bid Value 過去50個ticks 的平均大於現在的Bid Value 時，代表現在市場處於空頭，與報酬呈現負相關。

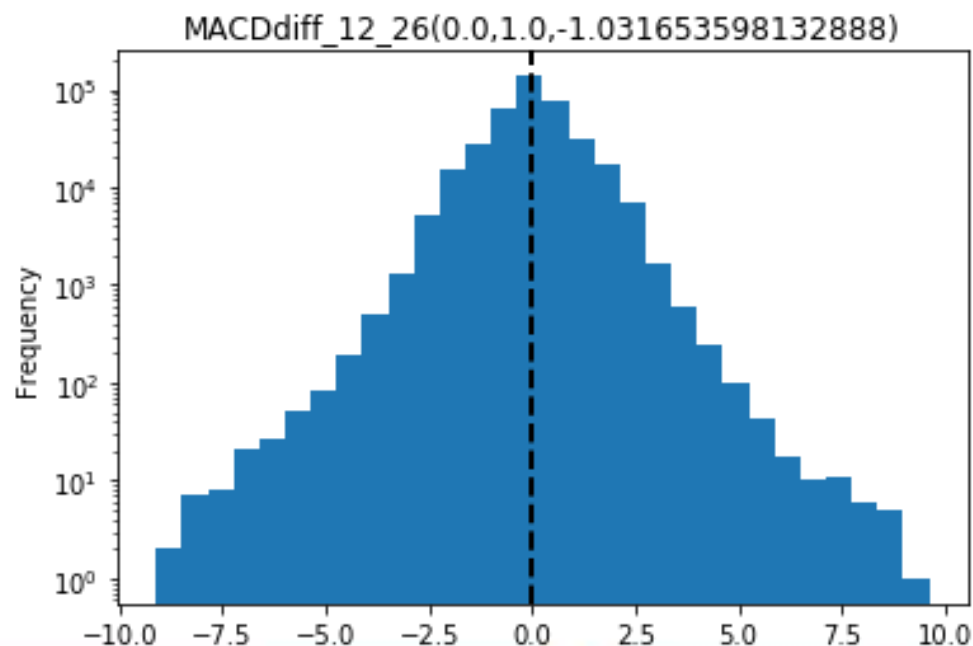
結合lasso模型: 因子

MACD

- DIF
短EMA (12個ticks) – 長EMA (26個ticks)

- MACD

DIF – DIF去取指數移動平均 (9個ticks)



因子與報酬的主觀解釋

- DIF: DIF 代表短 EMA – 長EMA。當短EMA – 長EMA 大於0時，通常表示**多頭**，因此報酬為正，與報酬為**正相關**。
- MACD: MACD大部分只有在黃金交叉以及死亡交叉時有用。因為當黃金交叉或死亡交叉後一段時間，會有**均值回歸**的現象。因此MACD才會是與報酬呈現**負相關**。舉例來說，當快線由下往上穿越時，也就是MACD剛從負變成正的時候，通常價格會上漲，但是經過一段時間後會產生均值回歸，價格下跌。

結合lasso模型: 因子

Spread of Value

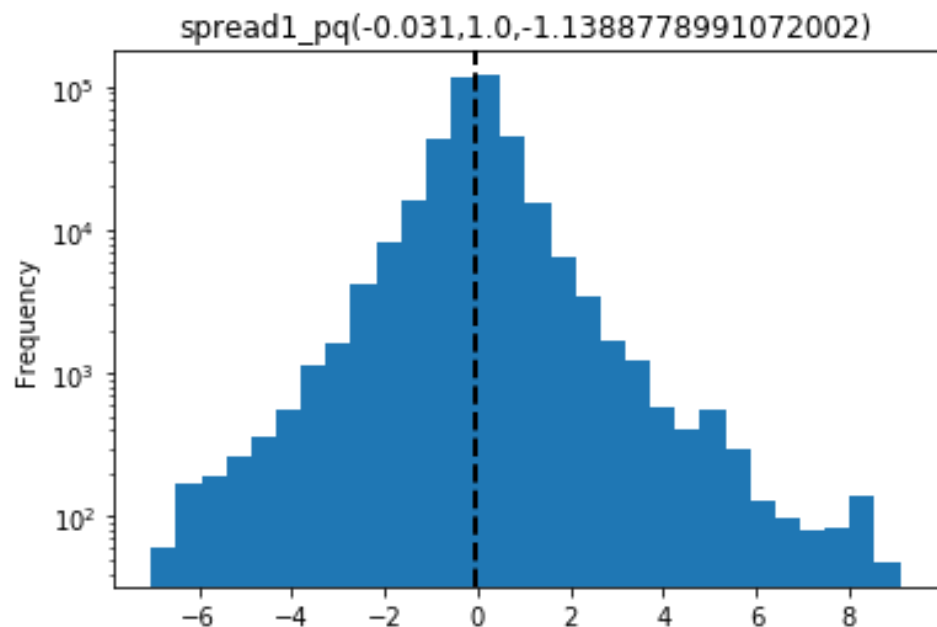
- Spread1 ~ Spread5
Ask Value – Bid Value

因子與報酬的主觀解釋

- Spread of Value: 負相關

因子與報酬的主觀解釋

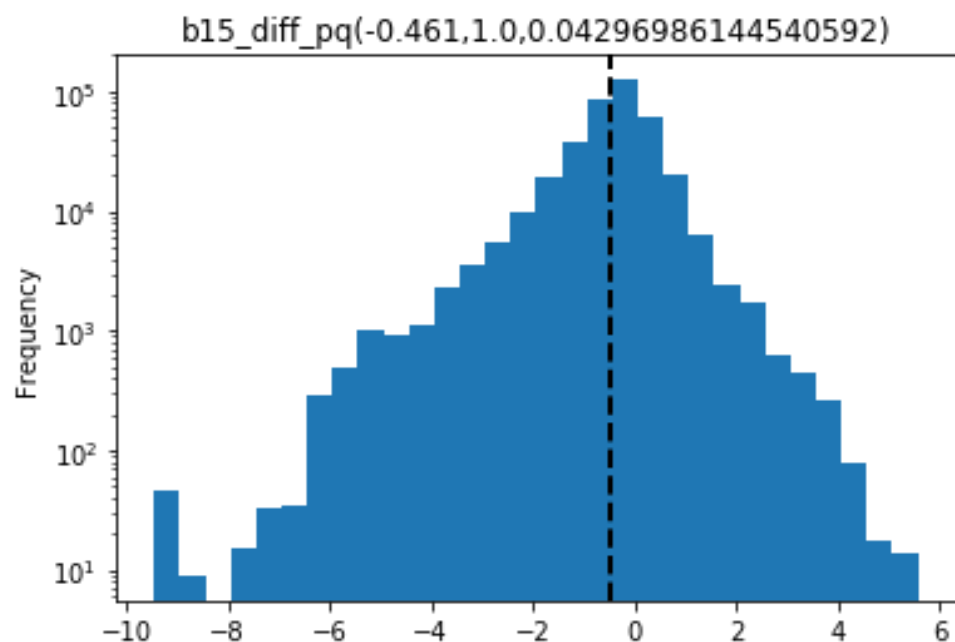
- Spread of Value: 我預期這群因子應該要與報酬呈現負相關，因為 Value 分別代表著買賣邊的力道。假如 Ask Value 大於 Bid Value，代表市場處於空頭，帶來的報酬為負。



結合lasso模型：因子

Value Differences

- Ask1 ~ Ask5 Value 之間的差
- Bid1 ~ Bid5 Value 之間的差



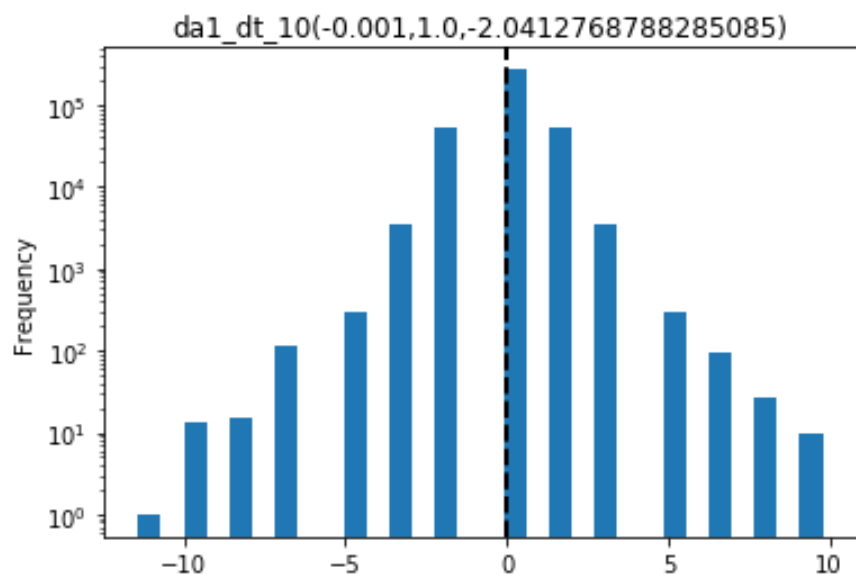
因子與報酬的主觀解釋

- Value Difference: 這群因子分別是Ask Value 之間的差以及Bid Value 之間的差。這群因子代表著5檔價量之間力道的差異。
- 舉例來說，假如Ask1 的力道遠大於Ask5 的力道，那代表現在市場上可能看跌，大家搶著要賣出，因此我認為這個因子與報酬是**負相關**。
- 反過來看，假如Bid1 的力道遠大於Bid5 的力道，那代表市場上現在想要逢低買進，未來趨勢看漲，因此我認為這個因子與報酬是**正相關**。

結合lasso模型: 因子

5 and 10 ticks 價取差分

- 5 & 10 ticks Ask 價取差分
- 5 & 10 ticks Bid 價取差分



因子與報酬的主觀解釋

- 5 & 10 ticks Ask 價取差分: 正相關
- 5 & 10 ticks Bid 價取差分: 正相關

我預期這群價格去取差分的因子應該要與報酬呈現**正相關**，因為我認為價格取差分，無論是Bid 還是 Ask 帶來的應該會是**動能**。也就是說如果過去10 個ticks 價格被trade上來，那下10個ticks 之後的報酬應該也會往上。

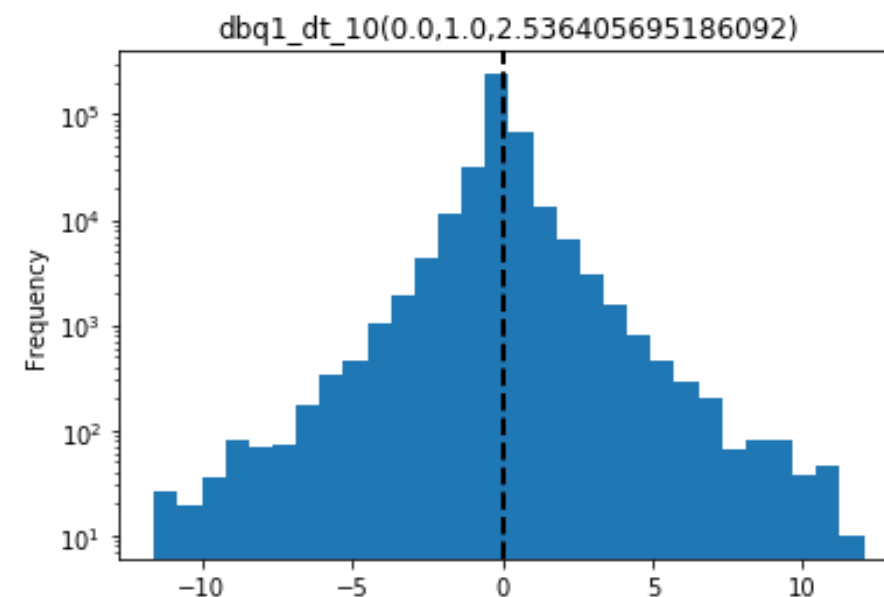
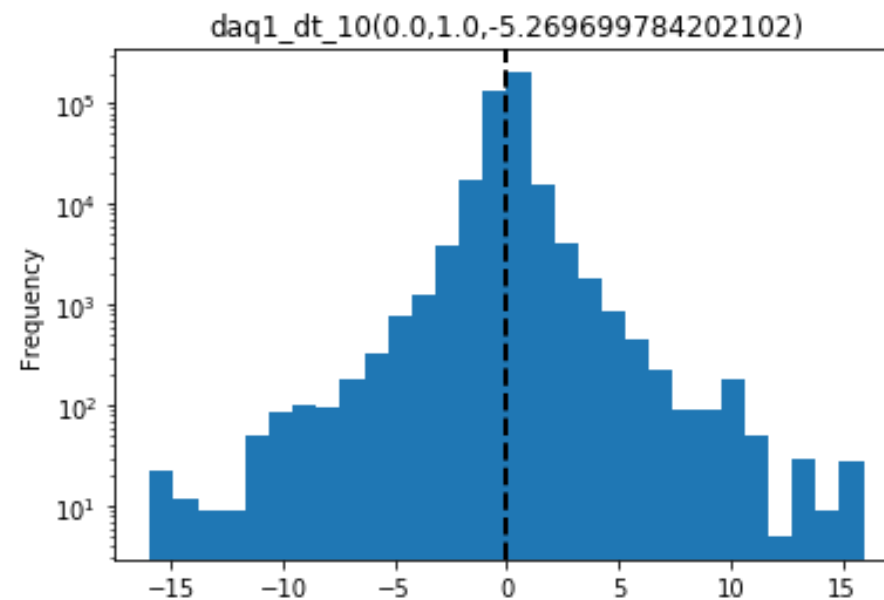
結合lasso模型: 因子

5 and 10 ticks 量取差分

- 5 & 10 ticks Ask 量取差分
- 5 & 10 ticks Bid 量取差分

因子與報酬的主觀解釋

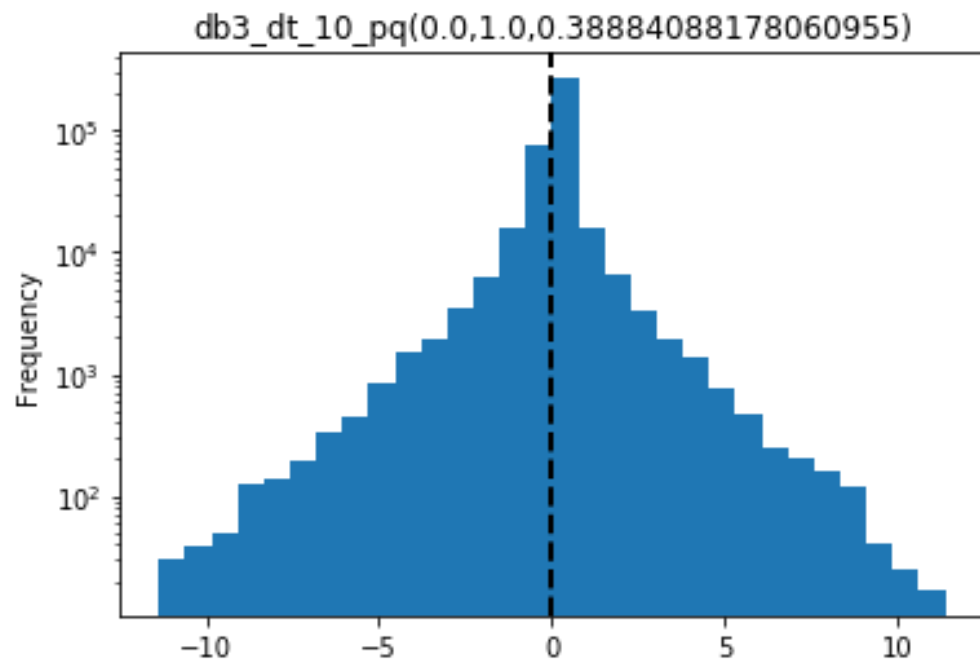
- 5 & 10 ticks Ask 量取差分: 我預期Ask的量取差分應該要與報酬呈現**負相關**，因為Ask的量取差分的值愈高，市場應該是**空頭**，帶來的是負的報酬。
- 5 & 10 ticks Bid 量取差分: Bid的量取差分應該要與報酬呈現**正相關**，因為Bid的量取差分的值愈高，市場應該是**多頭**，帶來的是正的報酬。



結合lasso模型: 因子

5 and 10 ticks Value 取差分

- 5 & 10 ticks Ask Value 取差分
- 5 & 10 ticks Bid Value 取差分



因子與報酬的主觀解釋

- 5 & 10 ticks Ask Value 取差分: 我預期Ask Value取差分應該要與報酬呈現**負相關**，因為Ask Value代表**賣邊的力道**，Ask Value與前10個 ticks 取差分的值愈高，應該代表市場空頭，股價報酬會往下。
- 5 & 10 ticks Bid Value 取差分: 我預期Bid Value 取差分應該要與報酬呈現**正相關**，因為Bid Value代表**買邊的力道**，Bid Value與前10個 ticks 取差分的值愈高，應該代表市場多頭，股價報酬會往上。

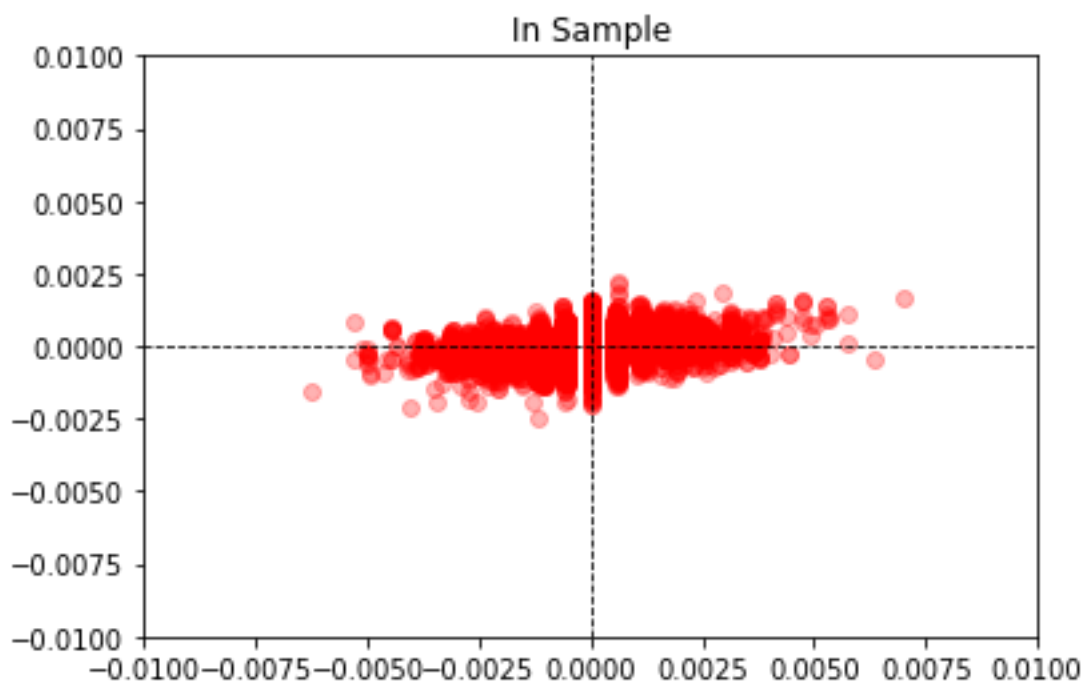
結合lasso模型: 重要因子

<i>factor</i>	<i>R-squared</i>	<i>factor</i>	<i>R-squared</i>
MA	0.068	5檔價取10個ticks 的 差分	0.018
MACD	0.02	5檔價取5個ticks 量 的差分	0.028
Spread of Value	0.039	5檔價取10個ticks 量 的差分	0.039
Value Differences	0.02	5檔Value 取5個ticks 的差分	0.028
5檔價取5個ticks 的 差分	0.019	5檔Value 取10個 ticks 的差分	0.038

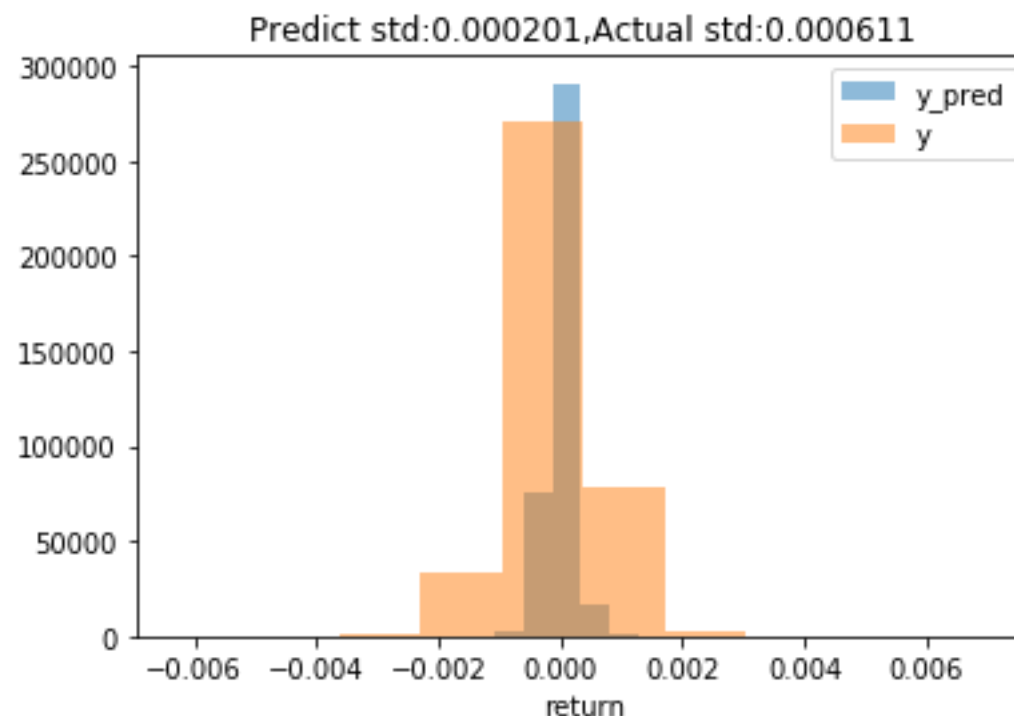
結合lasso模型: 模型預測下10個ticks的股價報酬 (3406 train)

Train R-squared: 0.11

真實的報酬與預測的報酬的散布圖



真實的報酬與預測的報酬的直方圖



結合lasso模型: 重要因子

<i>factor</i>	<i>Coefficient</i>
Ask1 Value 取10個ticks 差分	4.19
Ask2 Price 取10個ticks 差分	3.12
Bid1 取5個 ticks 量的差分	2.35
Bid2 取10個 ticks 量的差分	2.09
Bid1 取10個 ticks 量的差分	1.82

<i>factor</i>	<i>Coefficient</i>
Ask1 取10個 ticks 量的差分	- 4.66
Bid1 Value 取5個ticks 差分	- 2.64
Value1 的 spread	- 2.59
Ask1 Price 取10個ticks 差分	- 2.40
Bid1 Price 取10個ticks 差分	- 1.94

重要因子發現

- 模型選出的重要因子幾乎都是由**Ask1 或Bid1**所組成。這代表愈接近中間的價可以帶來愈多資訊，是符合預期的。
- 再者可以發現10個ticks 去取差分的效果比5個ticks 去取差分來的效果好。這也算是符合預期的，因為5個ticks 有時候可能擷取太短，發生價量都不太會動的情形，能夠提供的訊息量也較少。

結合lasso模型: 重要因子

10 ticks Ask Value 取差分

- $da1_dt_10_pq \sim da5_dt_10_pq$

因子與報酬的主觀解釋

- 10 ticks Ask Value 取差分: 負相關

模型因子係數

- $da1_dt_10_pq: + 4.19$
- $da2_dt_10_pq: + 0.32$
- $da3_dt_10_pq: - 1.55$
- $da4_dt_10_pq: - 0.24$
- $da5_dt_10_pq: + 0$

因子係數分析

- 原本預期這群因子應該要與報酬呈現**負相關**，因為Value已經考慮了價與量，我認為Ask Value與前10個 ticks 取差分的值愈高，應該代表**市場空頭**。
- 然而這個模型給這群因子的係數**偏正**，代表這群因子其實是代表著動能的。不過也有可能是因為這群因子捕捉的空頭資訊已經反應完了，下10個ticks **均值回歸**後反應的是**正的報酬**。

結合lasso模型: 重要因子

10 ticks Bid Value 取差分

- $db1_dt_10_pq \sim db5_dt_10_pq$

因子與報酬的主觀解釋

- 10 ticks Bid Value 取差分: 正相關

模型因子係數

- $db1_dt_10_pq$: - 1.34
- $db2_dt_10_pq$: - 1.51
- $db3_dt_10_pq$: + 0.35
- $db4_dt_10_pq$: + 0.24
- $db5_dt_10_pq$: + 0.23

因子係數分析

- 原本預期這群因子應該要與報酬呈現**正相關**，因為Value已經考慮了價與量，我認為Bid Value與前10個 ticks 取差分的值愈高，應該代表**市場多頭**。
- 然而這個模型給這群因子的係數**偏負**，可能是因為這群因子捕捉的多頭資訊已經反應完了，下10個ticks **均值回歸**後反應的是**負的報酬**。

結合lasso模型: 重要因子

10 ticks Ask 價取差分

- da1_dt_10 ~ da5_dt_10

因子與報酬的主觀解釋

- 10 ticks Ask 價取差分: 正相關

模型因子係數

- da1_dt_10: - 2.40
- da2_dt_10: + 3.12
- da3_dt_10: + 0
- da4_dt_10: - 0.22
- da5_dt_10: + 0

因子係數分析

- 原本預期這群因子應該要與報酬呈現**正相關**，因為我認為價格取差分帶來的應該會是**動能**。
- 然而這個模型給Ask1與Ask2這兩個取差分的因子係數剛好一負一正，有可能剛好被抵銷了，因此這群因子所代的動能不是這麼明顯。

結合lasso模型: 重要因子

10 ticks Bid 價取差分

- db1_dt_10 ~ db5_dt_10

因子與報酬的主觀解釋

- 10 ticks Bid 價取差分: 正相關

模型因子係數

- db1_dt_10: - 1.88
- db2_dt_10: + 0.41
- db3_dt_10: + 1.63
- db4_dt_10: + 0.68
- db5_dt_10: + 0.08

因子係數分析

- 預期這群因子應該要與報酬呈現**正相關**，因為我認為價格取差分帶來的應該會是**動能**。
- 這個模型給 Bid1 價取差分的係數是**負**的，Bid2 ~ Bid5 價取差分的係數則是**正**的。不過綜合來看**並沒有很明顯的偏正**，因此我無法說這群因子所帶來顯著的動能。

結合lasso模型: 重要因子

5 and 10 ticks Bid 量取差分

- dbq1_dt_5 ~ dbq5_dt_5
- dbq1_dt_10 ~ dbq5_dt_10

因子與報酬的主觀解釋

- 5 and 10 ticks Bid 量取差分: 正相關

模型因子係數

- dbq1_dt_5: + 2.35 dbq1_dt_10: + 1.82
- dbq2_dt_5: + 1.65 dbq2_dt_10: + 2.09
- dbq3_dt_5: + 0 dbq3_dt_10: + 0
- dbq4_dt_5: + 0 dbq4_dt_10: + 0
- dbq5_dt_5: + 0 dbq5_dt_10: + 0

因子係數分析

- 預期這群因子應該要與報酬呈現**正相關**，因為我認為 Bid 的量取差分的值愈高，市場應該是**多頭**，帶來的是正的報酬。
- 確實做出來的無論是5或10個 ticks 的量取差分，係數都是**偏正**的，也應證了這群因子所代表的特性。

結合lasso模型: 重要因子

10 ticks Ask 量取差分

- daq1_dt_10 ~ daq5_dt_10

因子與報酬的主觀解釋

- 10 ticks Ask 量取差分: 負相關

模型因子係數

- daq1_dt_10: - 4.66
- daq2_dt_10: - 0.78
- daq3_dt_10: + 1.17
- daq4_dt_10: - 0.05
- daq5_dt_10: - 0.13

因子係數分析

- 預期這群因子應該要與報酬呈現負相關，因為我認為 Ask 的量取差分的值愈高，市場應該是空頭，帶來的是負的報酬。
- 確實做出來10個 ticks 的量取差分，係數偏負的，也應證了這群因子所代表的特性。

結合lasso模型: 重要因子

Spread of Value

- spread1_pq ~ spread5_pq

因子與報酬的主觀解釋

- Spread of Value: 負相關

模型因子係數

- spread1_pq: - 2.59
- spread2_pq: - 0.95
- spread3_pq: + 0
- spread4_pq: - 0.33
- spread5_pq: + 0

因子係數分析

- 預期這群因子應該要與報酬呈現負相關，因為 Value 分別代表著買賣邊的力道，假如 Ask Value 大於 Bid Value，代表市場處於空頭，帶來的報酬為負。
- 確實做出來 Value 的 spread 係數是偏負的，也應證了這群因子所代表的特性。

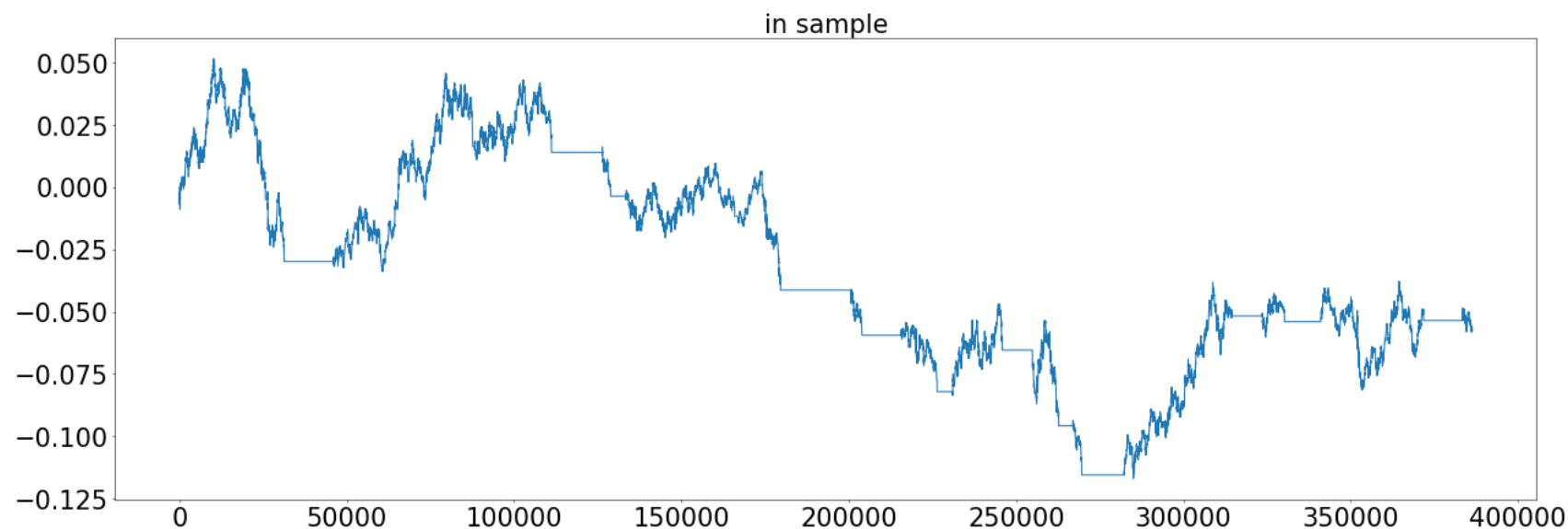
結合lasso模型: 模型回測結果 (3406 in sample)

Parameters

1. 預測未來 10個tick 漲 0.1% 做多
2. 預測未來 10個tick 跌 0.1% 做空
3. 虧 1.5%停損

Performance of train set

- Return: -6%
- Vol: 0.04
- Sharp Ratio: -1.62
- Transactions: 80
- Odds Ratio: 0.39
- Max Drawdown: 0.15



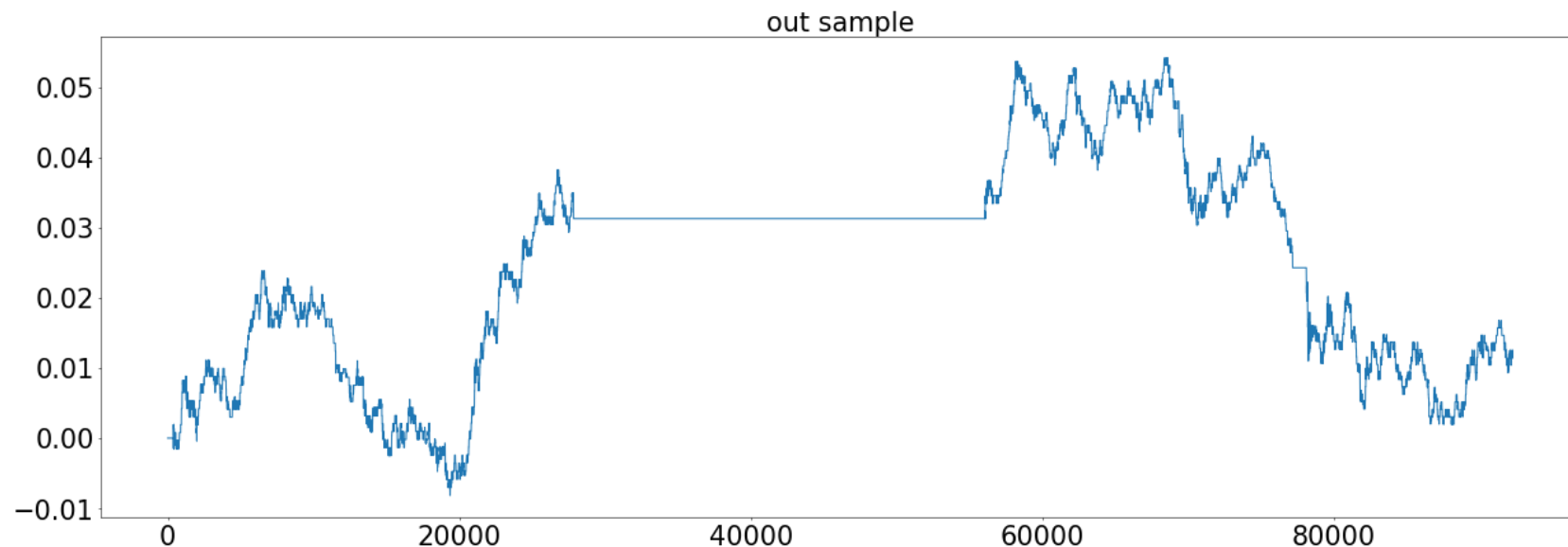
結合lasso模型: 模型回測結果 (3406 out sample)

Parameters

1. 預測未來 10個tick 漲 0.1% 做多
2. 預測未來 10個tick 跌 0.1% 做空
3. 虧 1.5%停損

Performance of train set

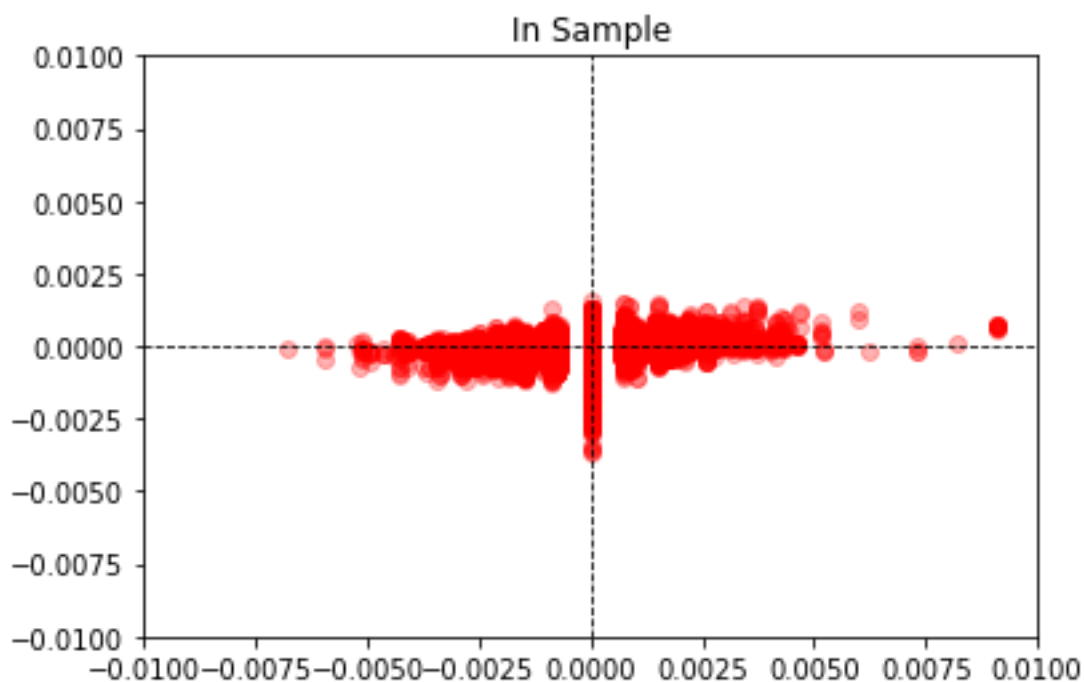
- Return: + 1.1%
- Vol: 0.014
- Sharp Ratio: 0.42
- Transactions: 18
- Odds Ratio: 0.28
- Max Drawdown: 0.04



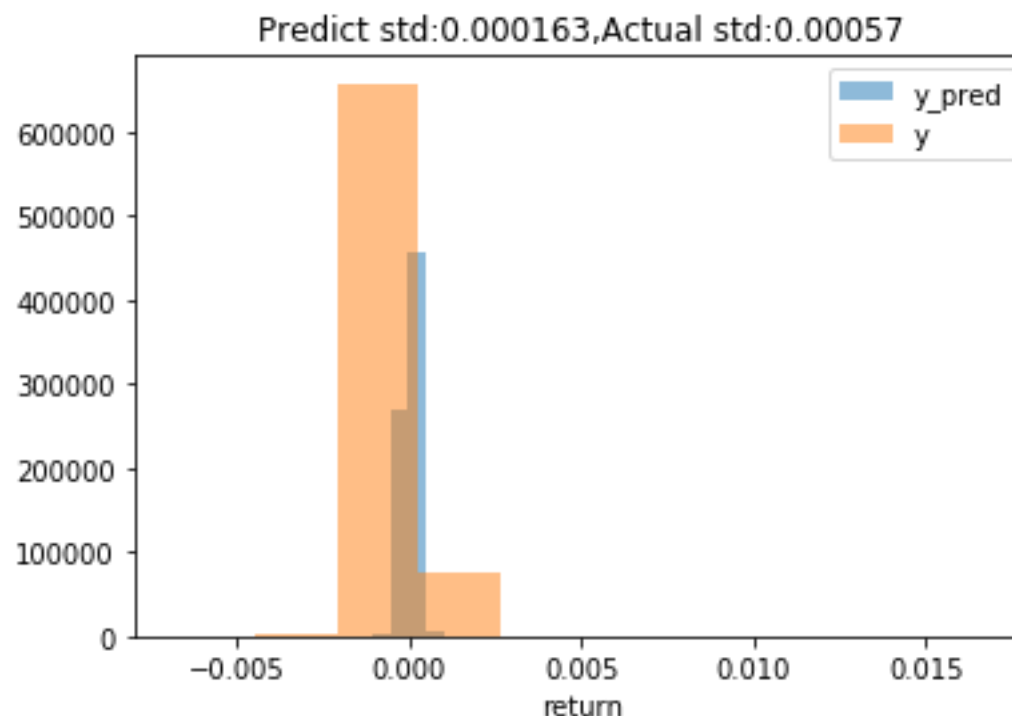
結合lasso模型: 模型預測下10個ticks的股價報酬 (2327 train)

Train R-squared: 0.085

真實的報酬與預測的報酬的散布圖



真實的報酬與預測的報酬的直方圖



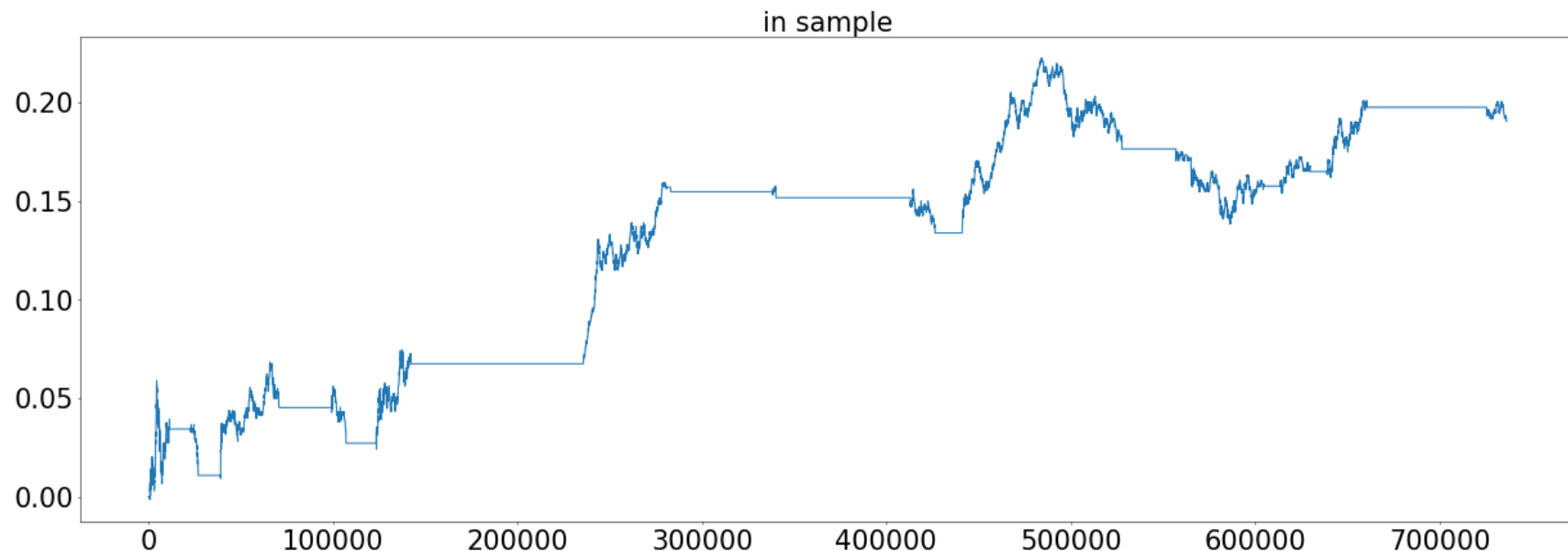
結合lasso模型: 模型回測結果 (2327 in sample)

Parameters

1. 預測未來 10個tick 漲 0.1% 做多
2. 預測未來 10個tick 跌 0.1% 做空
3. 虧 1.5%停損

Performance of train set

- Return: **18.7%**
- Vol: 0.05
- Sharp Ratio: **3.17**
- Transactions: 35
- Odds Ratio: **0.57**
- Max Drawdown: 0.03



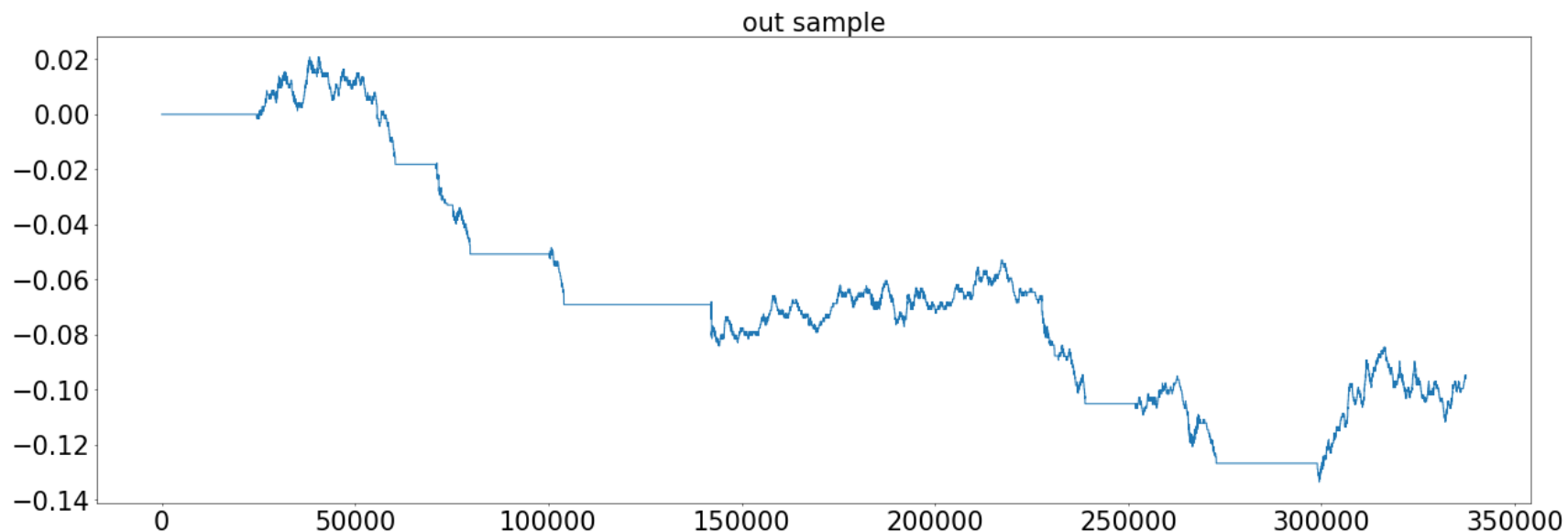
結合lasso模型: 模型回測結果 (2327 out sample)

Parameters

1. 預測未來 10個tick 漲 0.1% 做多
2. 預測未來 10個tick 跌 0.1% 做空
3. 虧 1.5%停損

Performance of train set

- Return: -9.7%
- Vol: 0.43
- Sharp Ratio: -2.38
- Transactions: 21
- Odds Ratio: 0.14
- Max Drawdown: 0.13



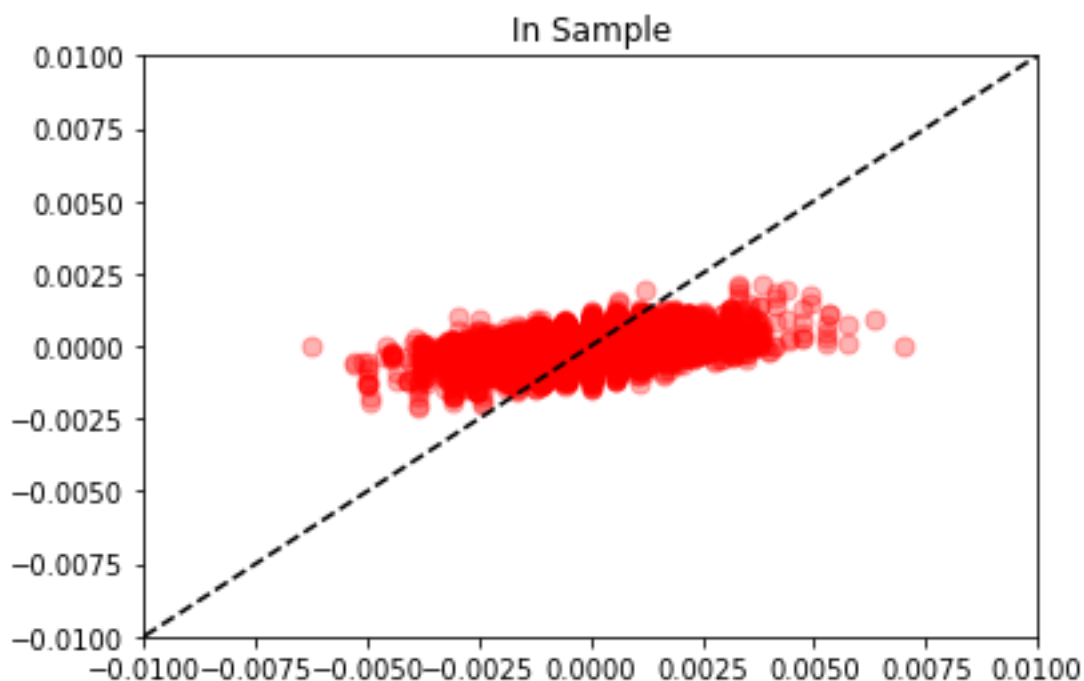
結合lasso模型: 結合LSTM



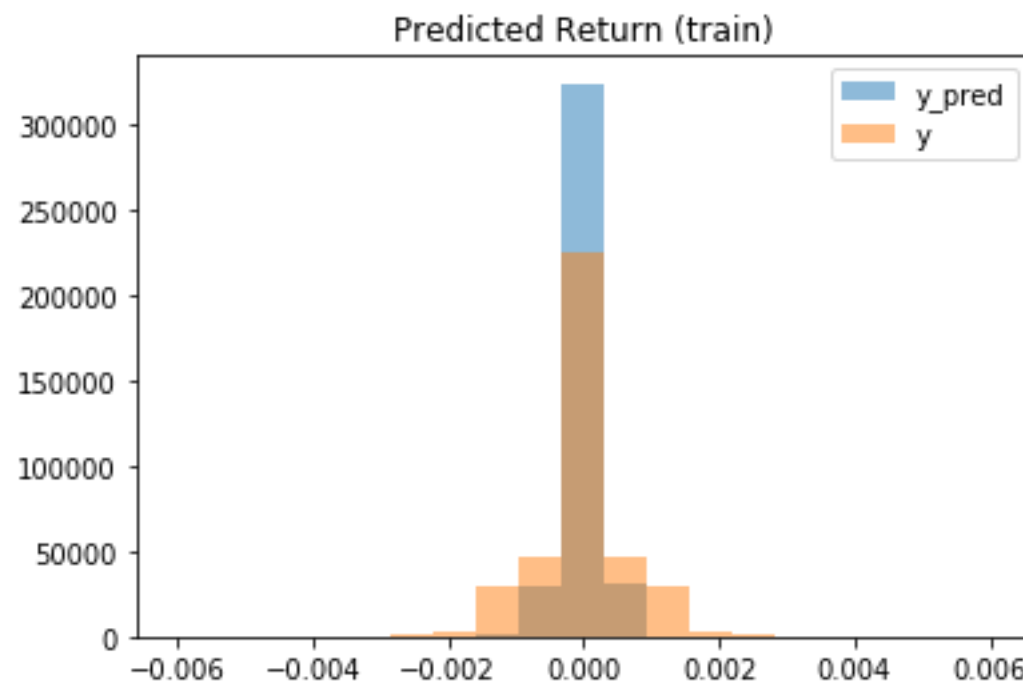
結合lasso模型: LSTM 預測下10個ticks的股價報酬 (3406 train)

Train R-squared: 0.176

真實的報酬與預測的報酬的散布圖



真實的報酬與預測的報酬的直方圖



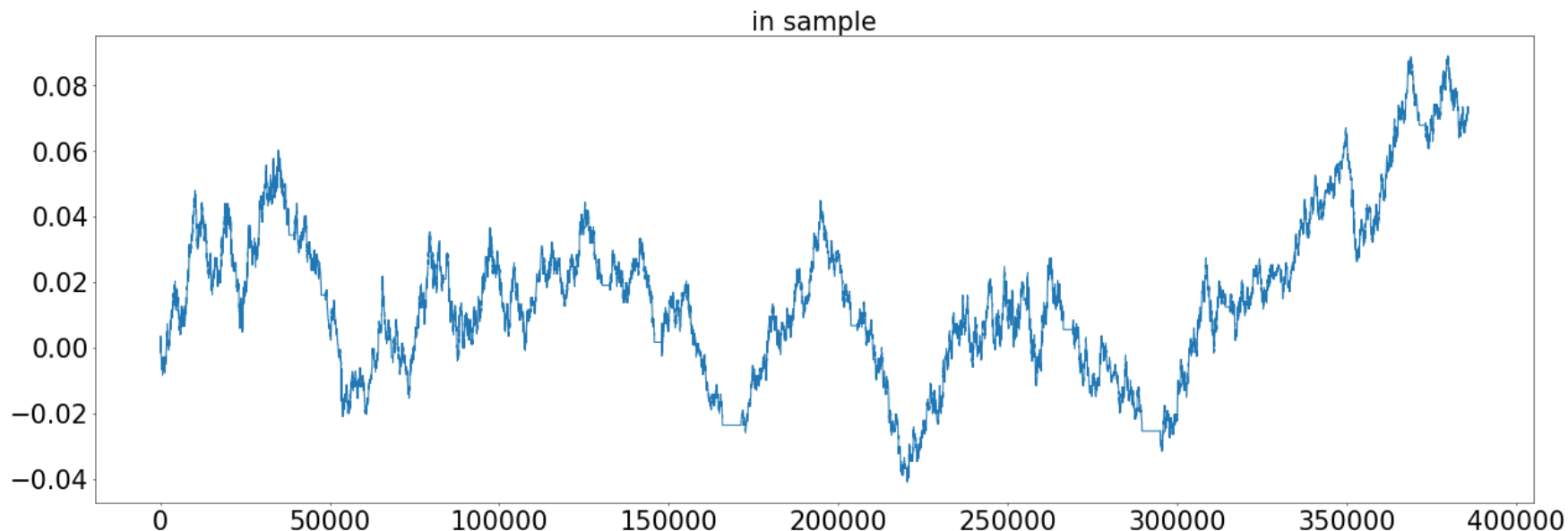
結合lasso模型: LSTM 回測結果 (3406 in sample)

Parameters

1. 預測未來 10個tick 漲 0.1% 做多
2. 預測未來 10個tick 跌 0.08% 做空
3. 虧 1.5%停損

Performance of train set

- Return: 7.1%
- Vol: 0.023
- Sharp Ratio: 2.83
- Transactions: 87
- Odds Ratio: 0.46
- Max Drawdown: 0.089



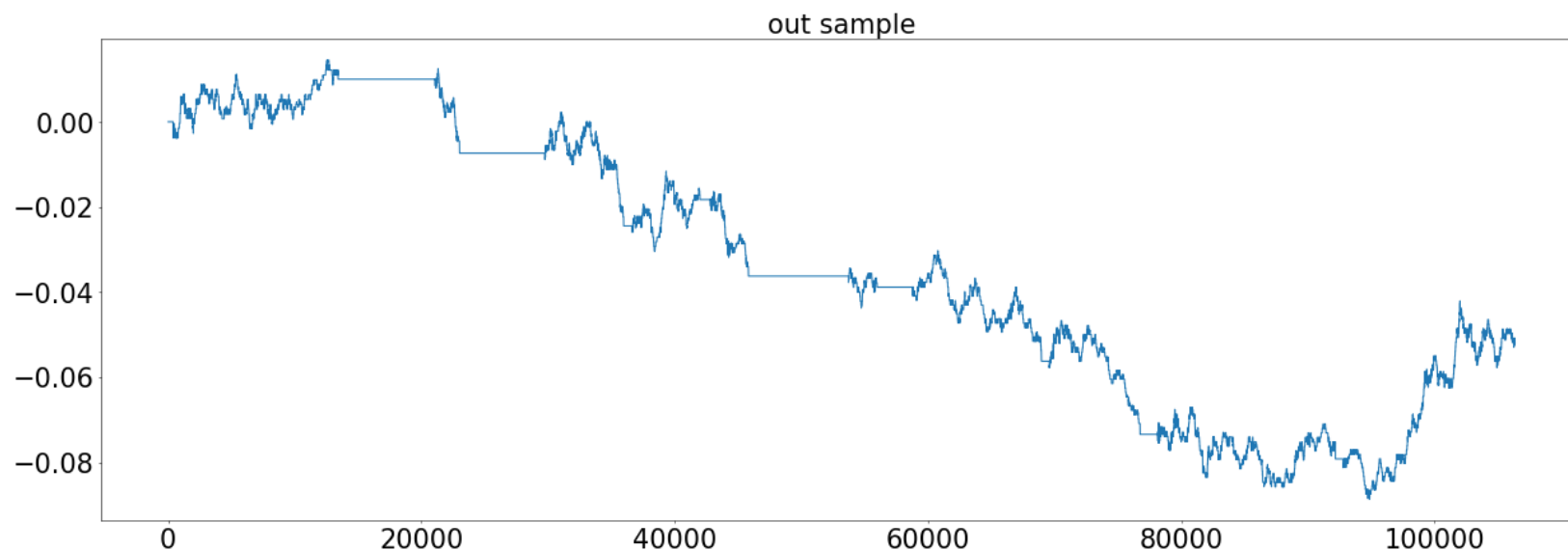
結合lasso模型: LSTM 回測結果 (3406 out sample)

Parameters

1. 預測未來 10個tick 漲 0.1% 做多
2. 預測未來 10個tick 跌 0.08% 做空
3. 虧 1.5%停損

Performance of train set

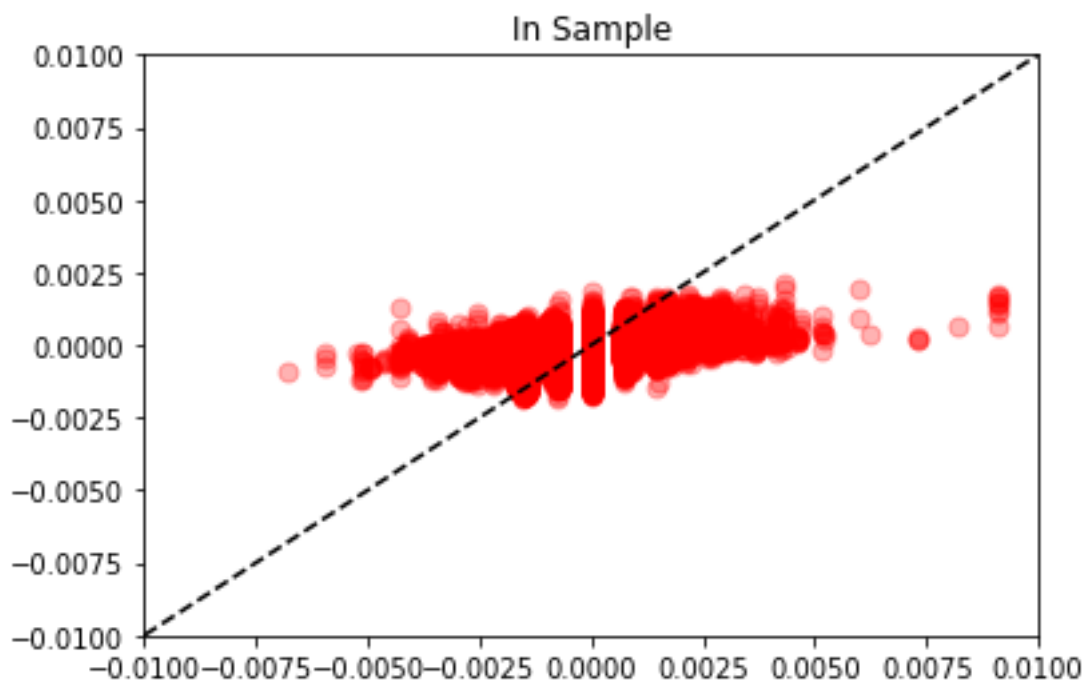
- Return: -5.4%
- Vol: 0.031
- Sharp Ratio: -1.92
- Transactions: 10
- Odds Ratio: 0.3
- Max Drawdown: 0.09



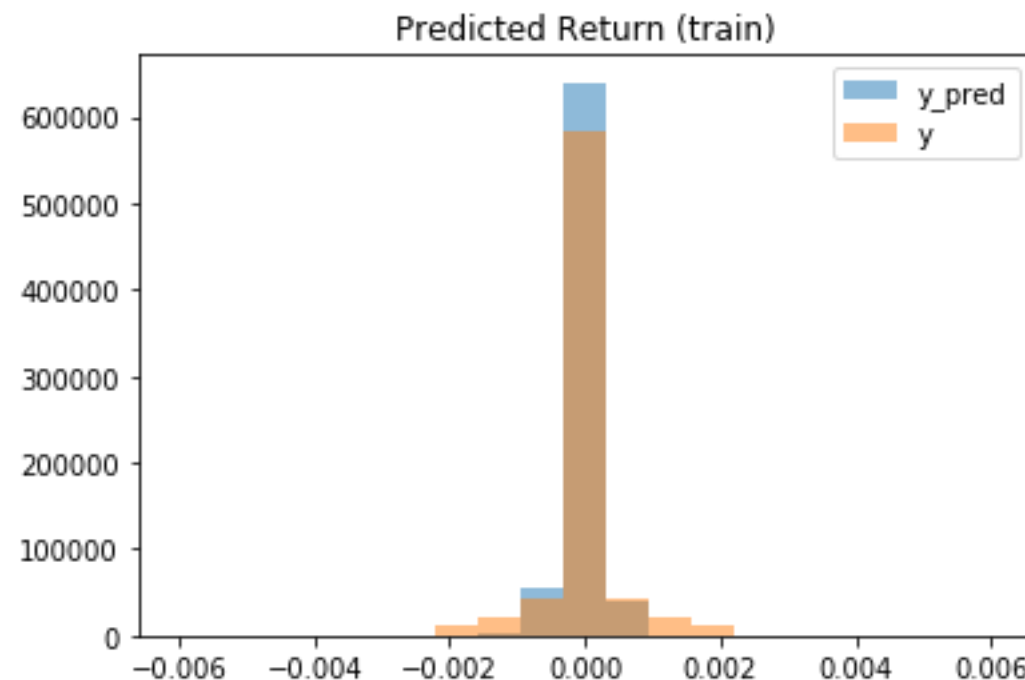
結合lasso模型: LSTM 預測下10個ticks的股價報酬 (2327 train)

Train R-squared: 0.15

真實的報酬與預測的報酬的散布圖



真實的報酬與預測的報酬的直方圖



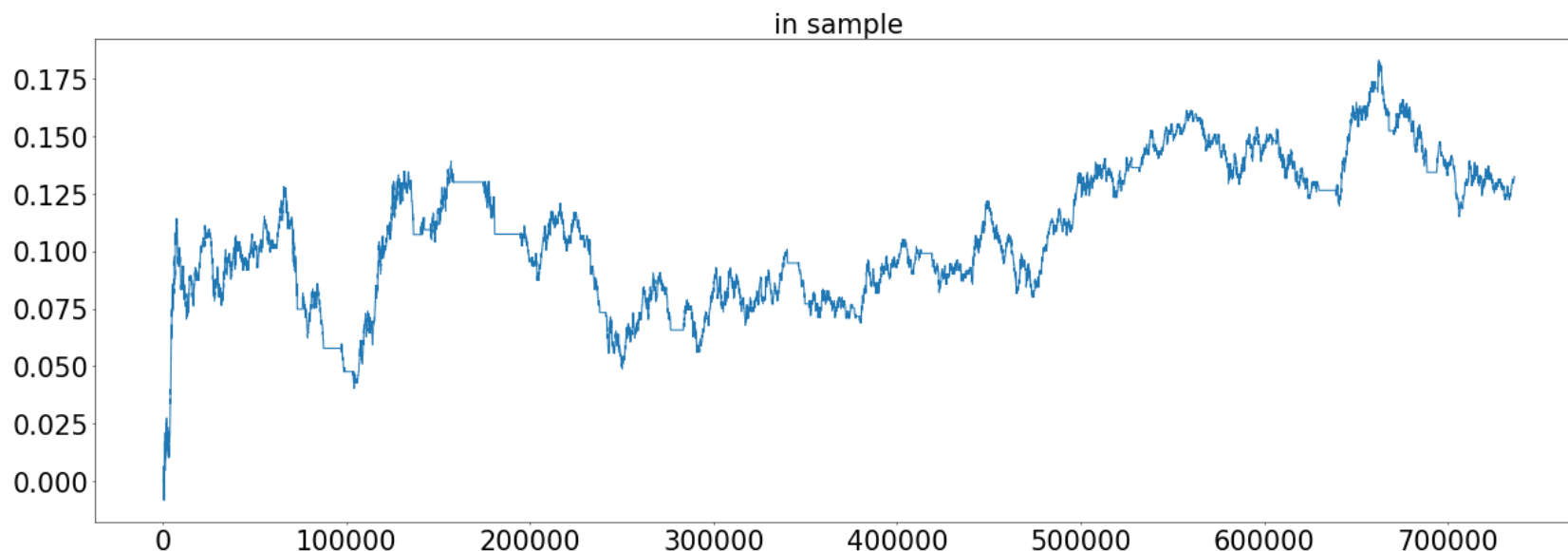
結合lasso模型: LSTM 回測結果 (2327 in sample)

Parameters

1. 預測未來 10個tick 漲 0.1% 做多
2. 預測未來 10個tick 跌 0.1% 做空
3. 虧 1.5%停損

Performance of train set

- Return: **13.2%**
- Vol: 0.031
- Sharp Ratio: **4.14**
- Transactions: 106
- Odds Ratio: **0.53**
- Max Drawdown: 0.074



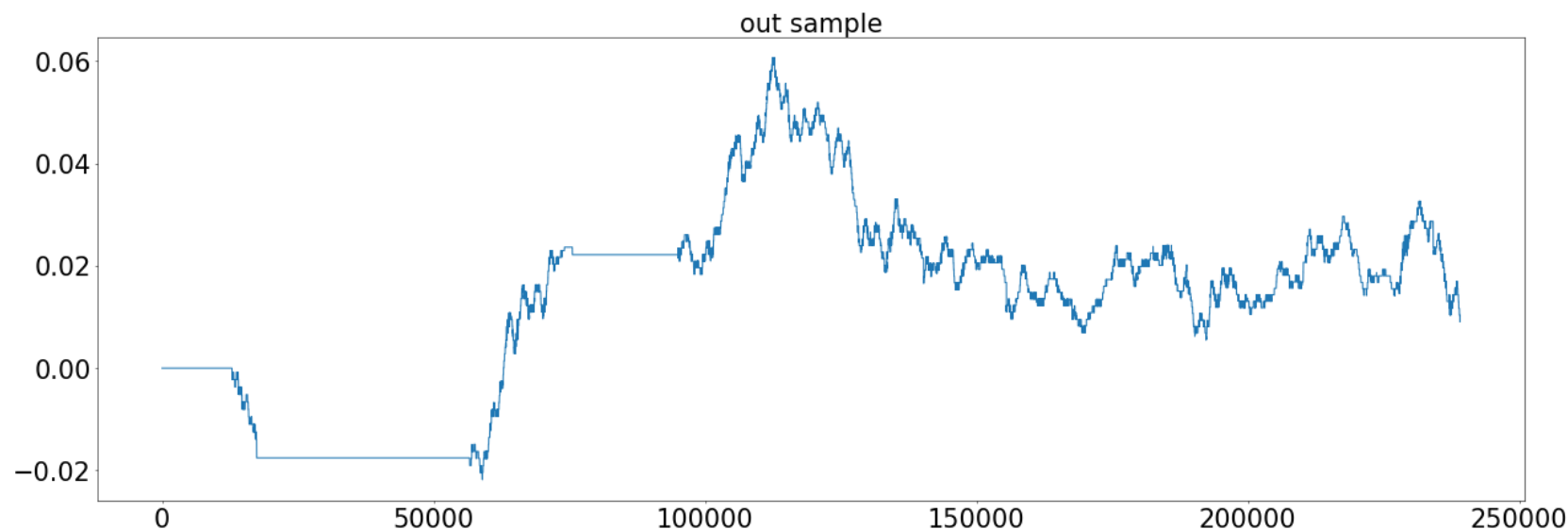
結合lasso模型: LSTM 回測結果 (2327 out sample)

Parameters

1. 預測未來 10個tick 漲 0.1% 做多
2. 預測未來 10個tick 跌 0.1% 做空
3. 虧 1.5%停損

Performance of train set

- Return: 0.77%
- Vol: 0.013
- Sharp Ratio: 0.183
- Transactions: 11
- Odds Ratio: 0.36
- Max Drawdown: 0.022



結合學lasso模型: 小結

1. 深度學習預測出來的報酬，在丟入相同的因子的情況下，經由回測結果顯示，績效比用較簡單的回歸模型好。不過深度學習模型還是需要回歸模型輔助選取因子，除了可以大大提升模型的訓練效率之外，預測的準確度也有所提升。
2. 初步完成這次預測個股報酬的研究後發現，最重要的還是因子的開發。然而目前較缺乏交易的經驗以及理論的背景，尚未開發出更有用、複雜的因子，希望日後能夠累積經驗，藉由交易的直覺去開發出新的因子。

未來研究

- 結合BERT模型挖取新聞資訊對股價報酬的影響
(參考論文: BERT-based Financial Sentiment Index and LSTM-based Stock Return Predictability)
- 持續挖掘新的因子
(閱讀國內外最新論文，擷取新的因子想法)
- 改善模型普遍低估股價報酬的情形
(嘗試對股價報酬做一些轉換，讓預測值不要只偏好貼在0上)

謝謝聆聽