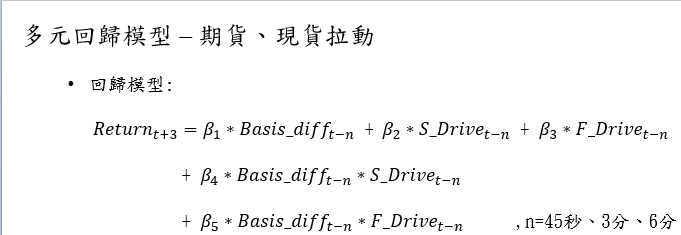
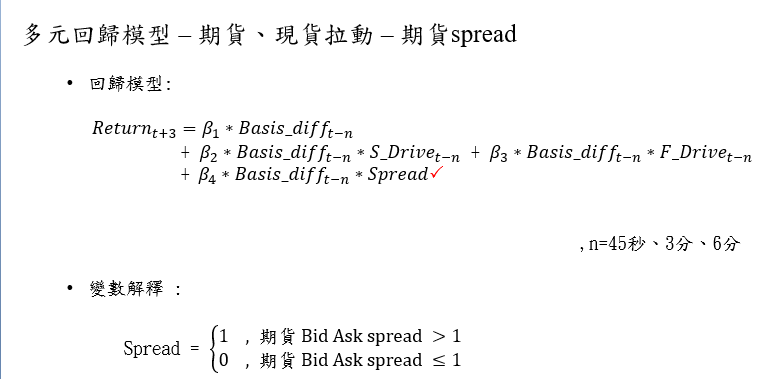
2/24



簡單線性回歸 -> 一個一個挑進來 (為何不先挑一個，接著2.3.4…….)

共線性問題? Why use linear regression?

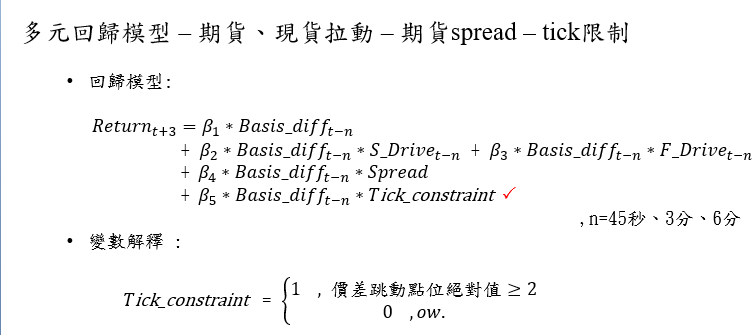
現貨拉動? 期貨拉動? S\_Drive、F\_Drive



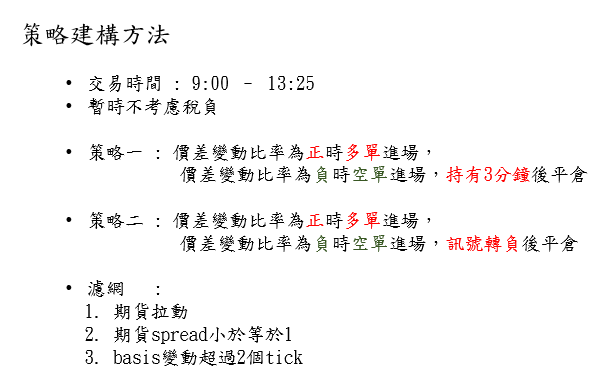
期貨spread? 為何要納入這個指標

Q -> 訊號出現後，我們假設可以在下一筆成交資料進場，如果**進場當下期貨spread變寬**，則會**現賠變寬的點數?**

P16 -> basis只變動一個tick時，報酬分配較為分散，**容易頻繁進場又出場**；而tick = 2時，75%以上報酬為正，tick = -2時，75%報酬為負 -> 設定一個跳動區間為 2~7



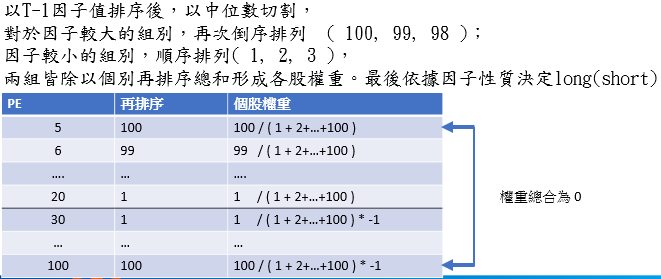
整理好重要因子後(**期貨拉動、期貨spread <= 1、basis變動 > 2 tick**) -> 開始回測



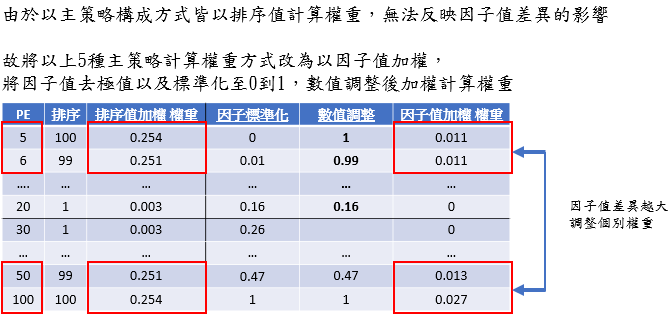
專案主題3：日間因子投組研究

主策略1 -> 5種

主策略1-3 排序權重法 -> 意義? (重新排序權重，更標準?)

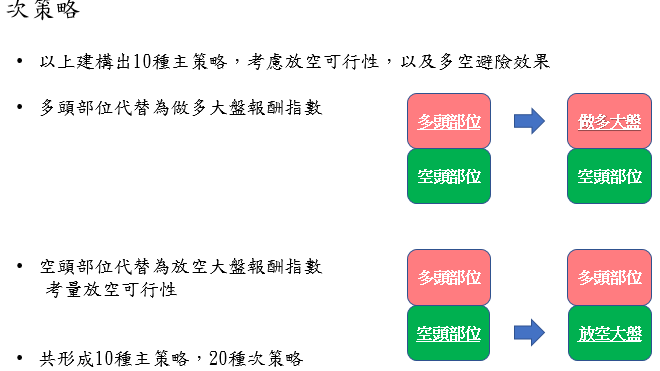


主策略2 因子值加權 (策略1的5種)



Total: 10種主策略

次策略



Total: 20種次策略(多頭 -> 做多大盤；空頭 -> 放空大盤)

2/25

門密碼

12324565

Data

基本資料：**成交量、收盤價、總市值、成交金額**

Code錯誤改正 -> df\_V\_melt -> df\_V\_melt\_new (line47)

產業數據：**產業代號、台灣發行量加權股價報酬指數(大盤指數)**

<https://whhnote.blogspot.com/2011/01/python-strtime-strftime.html>

(時間格式轉換方法)

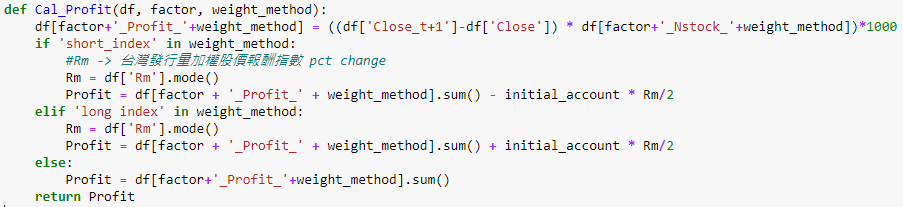
[datetime.strftime(x, format='%Y%m%d') for x in df\_RM['Date'] ]

df\_merge1\_new更正

CMoney data：**PE、PB、外資買賣超(for)、券資比(rgz)、外資買賣超金額(for\_N)、投信買賣超(invest\_N)、主力買賣超張數、股本(sharecapital)**

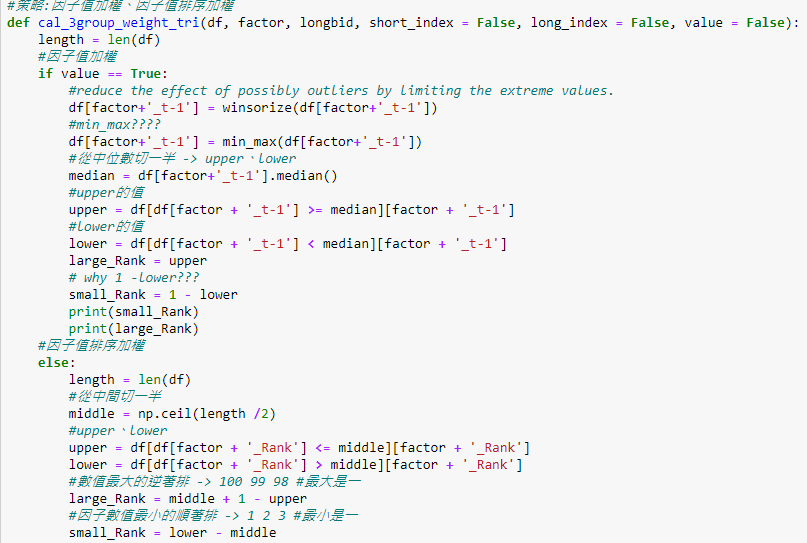
其他資料：**VWAP(成交量加權平均價)、SALES、div(股利)**

Df\_merge3\_new 更正



Df[‘RM’.mode ??、why initial account/2，分一半? -> 一半short/long大盤，一半long/short台股

因為是取每日資料，基本上RM只有一個值 -> 取眾數沒差 (solved)



**因子值加權**、**因子值排序加權**

2/26

策略想法

篩選：平均日成交金額 >= 5000萬 、以T-1日市值排序 -> 選取前200大個股(加上**挑選前10大產業**)

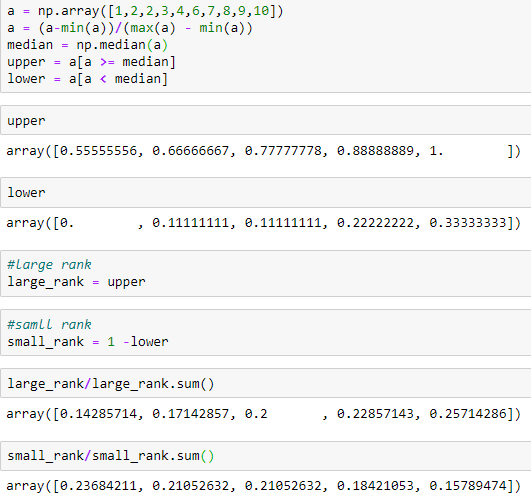
排序權重法(**考慮產業**) + 因子值加權 ex: PE ratio

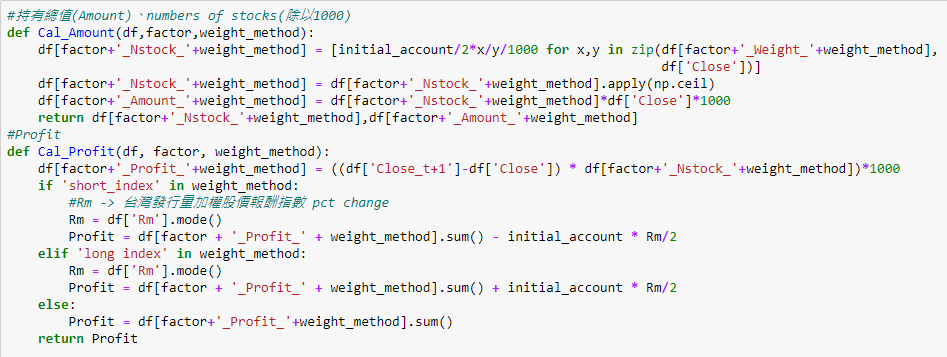
排序PE -> 因子去極值、標準化 -> 從中位數切一半

-> **Upper 、 lower** -> large rank = upper、lower rank = 1- lower

-> 依照策略計算weight

買(賣)因子大的，賣(買)因子小的 (一買一賣 -> **delta對消的概念**)

策略1 因子值加權法



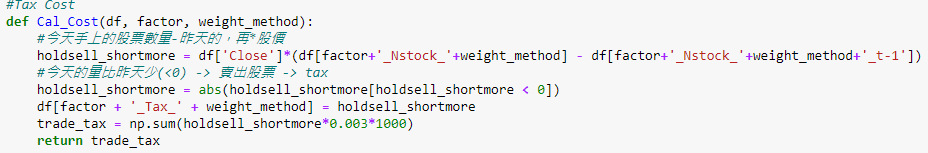
每天10億 -> 5億long/5億short -> initial amount / 2

問題：**沒有考慮到account剩下多少錢的問題**?

* 可能是只hold一天，每天進出場?
* 不過**隔天又會有initial account的一半的錢**可以花，有點奇怪

計算equity value的方法

最後initial value + **逐天累加的porfits** – 逐天累加的tax



依據前面的function(Cal\_Amount)來看，感覺每天的買賣是獨立的，因此以此為標準來判斷是否有賣出股票有點奇怪 (已解決)

Ex:昨天某張股票買300張，今天的算法得出要買200張 -> 賣出100張?

* 今天和昨天的價差去計算損益(profits)，昨天20今天21，if long -> 賺1\*N\_stocks
* 最後逐天累加，再加上初始金額 -> 最後的總equity

進度:line372 strategy 2

3/2

篩選台股標準

1. 以平均日成交金額5000萬為篩選標準
2. 總市值前200大

* temp\_m

1. 選出前10大產業產業(數量前10多的)

* temp\_i

1. 平均權重法 考慮產業 -> Ind\_CAP\_Weight

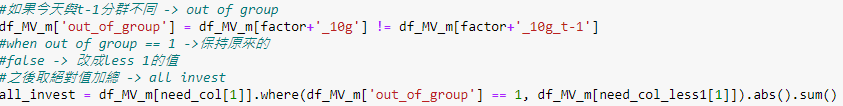
策略2

**問題**

1. 不懂取out of group的意思

Out of group -> 昨天與今天的分群不同 (ex:PE變化大時)

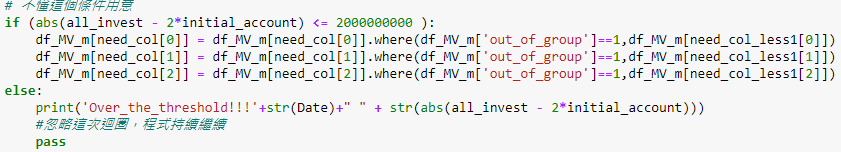
* 如果PE變化過大時，仍然買昨天的amount



1. |all invest – 2 \* initial value| <= 2 \* initial value (20億)

差異過大時(t、t-1期分在不同群) -> 將t期改成t-1期的stocks、amounts、weight

Where(out of group == 1 -> 改成 less1)



策略1 因子值排序加權 (tri)

1. 挑選總市值前200大
2. 將t-1期的PE排序，給予排名 1~200
3. 給予權重

策略2 因子值排序加權 + 10組固定 (tri\_10g)

策略3 因子值排序加權 + 產業百分比(scale) (tri\_ind%)

1. 挑選總市值前200大
2. 將t-1期的PE排序
3. 將其去極值並標準化(scaling) -> PE\_t-1
4. 又因太多0、1 -> 依照PE\_t-1\_%排序後，如有相同，再依照原本的PE\_t-1去排序
5. 給予排名 1~ 200
6. 給予權重

策略4 因子值排序加權 + 產業百分比 + 10組固定 (tri\_ind%\_10g)

策略5 因子值排序加權 + 產業標準化 (tri\_indstd)

同策略3，唯獨標準化的方式為std

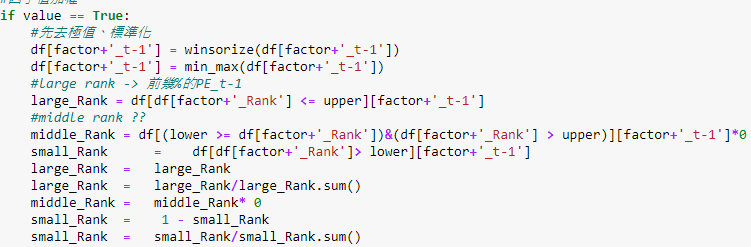
策略6因子值排序加權 + 產業標準化 + 10組固定 (tri\_indstd\_10g)

**back\_test\_multiway\_PE**

問題

1. middle\_Rank ??

* 這裡不配給權重，中間不作買賣 ex: PE買後10%賣前10%



3/3

計算權重(因子值加權->以因子值去作加權、因子值排序加權->以因子值的rank去作加權)

1. 排序權重法

2. 平均權重法

large rank、small rank ex. 取前後10%，並給予equal weight

main strategy

#因子值排序加權

1. 排序權重法(tri)

初始的function

1. 排序權重法 考慮前10大產業(tri\_ind)

依照各個產業類別分別去平均分配權重、排序rank

上述權重最後再除上本身產業的個數 -> 真正的權重 (總權重和=1)

1. 排序權重法 考慮前10大產業 考慮產業市值

依照各個產業類別分別去平均分配權重、排序rank

上述權重最後再\*先前算出的Ind\_CAP\_Weight(前10產業市值之權重)

1. 平均權重法(eq)
2. 平均權重法 考慮前10大產業(eq\_ind)

ex. long/short因子值最大的2檔個股，並short/long因子值最小的2檔個股

，並給予equal weight

上述權重最後再除上本身產業的個數 -> 真正的權重 (總權重和=1)

1. 以上結合放空指數or做多指數
2. 將上述全改成#因子值加權(value)

**Select factor by machine learning**

因子: 過去一年的 -> PE(取倒數)、PB(取倒數)、ROE、市值(取對數 ~ Normal)、3 months beta

Response: 下一個月的報酬

regression coefficient (factor return) is calculated by using the monthly factor value of each stock of the **past 1 year (before the t-phase) as the explanatory variable** and by using **the end of the next month (t-phase) return as the target variable**. We have calculated the future stock return rate (expected return) of each stock by **multiplying the resultant regression coefficient (factor return) with the factor value (factor weight) of each stock** at the end of the test period.

we considered **first quantile** of the five quantiles portfolio made on the basis of the resultant expected return as **long portfolio** and **fifth quantile** as **short portfolio**, and calculated the difference in return (spread return) of the long portfolio and the short portfolio when held

until the next month. We rolled this on a monthly basis and accumulated the obtained returns.

問題

Out of group那幾個策略，用t、t-1的資訊去買賣t期的股票有點奇怪

* 改 t-1、t-2 (自己看錯，solved)

作業

1. 預測一檔股票的中價 (bid+ask)/2 using machine learning algorithms

<https://www.groundai.com/project/machine-learning-for-forecasting-mid-price-movement-using-limit-order-book-data/1>

3/4

Machine learning for forecasting mid price movement using limit order (限價單) book data

* a limit order is a type of order to buy or sell a specific number of shares within a set price.

Data: financial time-series data

* inherently noise and non-stationary (不適合用統計模型 -> 因為常需要分配的假設等等), machine learning algorithms usually take no such distribution assumption.
* NN、SVM、deep learning methods (capable of modeling highly non-linear and complex data)

\*limit order data changes distribution frequently even within one day.

\*feature extractions (ex. NN、bag-of-features)

Three classifiers are evaluated using combinations of these sets of features on two different evaluation setups and three prediction scenarios.

Three scenarios are assessed regarding the span of time for which predictions are made: a scenario where (1) **the movement of the mid price of the immediately succeeding sample in time is predicted,** one where (2) **the average mid price movement of the next five samples is predicted** and, last, one where (3) **the average movement of the mid price for the next ten samples is predicted**.

\*Past work

Ex. Using NN SVM MLP -> using two diff windows, long and short term

and it is shown that **using too many samples** that span too far into the past can degrade the prediction quality.

In a similar fashion to this work, kercheval2015modelling uses several handcrafted **time sensitive and insensitive features**, extracted from the limit order book. These features include **bid-ask spreads** and **mid prices**, **price differences**, **mean prices** and **volumes**, along with derivatives of the price and volume, average and relative intensity indicators, totaling to 144 different features. However, in kercheval2015modelling the proposed methods are evaluated on a very small dataset that contains about 400,000 order book rows. In contrast, in our case a large-scale dataset that contains information for **10 days and 5 stocks** is used, with the raw data being more than **4 million samples**.

\*Features from high frequency limit order book

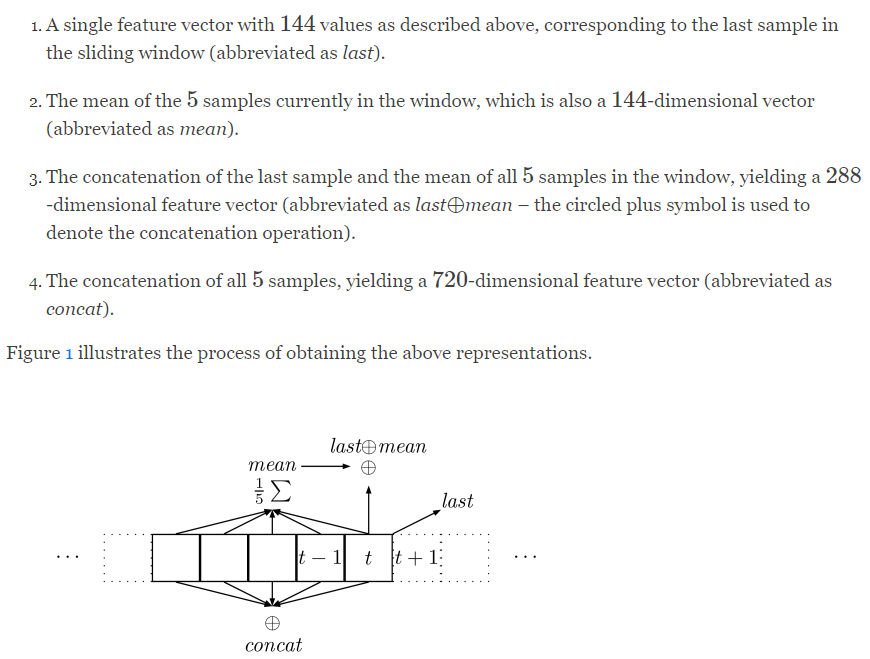
Bia、ask、bid volumes、ask\_volumes

The data used in this work consists of **10 orders for each side (bid, ask) of the LOB**. Each order is described by **2 values, the price and the volume**, yielding a **total of 40 values for each time step**. The time period used for collecting that data ranges from the 1st to the 14th June 2010 (only business days are included), and the data is provided by the Nasdaq Nordic data feeds siikanen2016limit (); siikanen2017drives (). The dataset is made up of

10 days for 5 different stocks and the total number of messages is about 4.5 million with equally many separate depths.

More specifically, first, a basic set of features which includes the **prices and volumes** for every level of t**he ask and bid side of the order book** is extracted. This information yields **40 values** at each time step. Then, time-insensitive features describing **the spread, mid-price, price and accumulated price differences between the bid and ask orders** of each depth level, and price and volume spreads are extracted. Finally, time-sensitive features are extracted corresponding to the **average intensity for trades, orders, cancellations, deletion, execution of visible limit orders and execution of hidden limit orders**. This set of features also includes **the price and volume average values at each level of the LOB, the average intensity per trading type as well as comparisons between the intensities and limit activity acceleration** (derivatives of average intensities). Because of the non-linear nature of time in the LOB data, we follow an event-based inflow.

Feature extracted methods

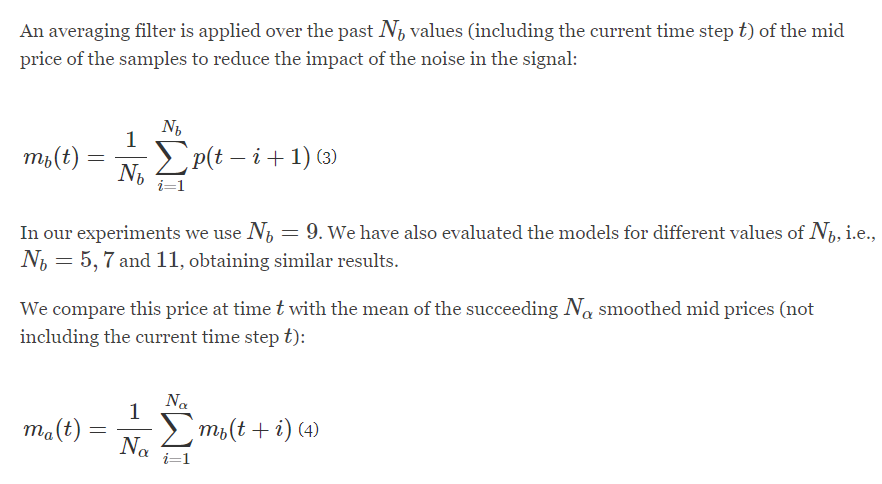


Using this sliding window approach allows for **temporal information to be incorporated** into the representations, which are subsequently used to predict the movement of the stock’s mid price. By averaging over the five entries contained in the window, some of the inherent noise of the features is alleviated. Concatenating (聯集) different representations, e.g., the last feature vector and the average of the last 5 feature vectors, allows for more temporal information to be introduced in the final feature vector.

Predition labels

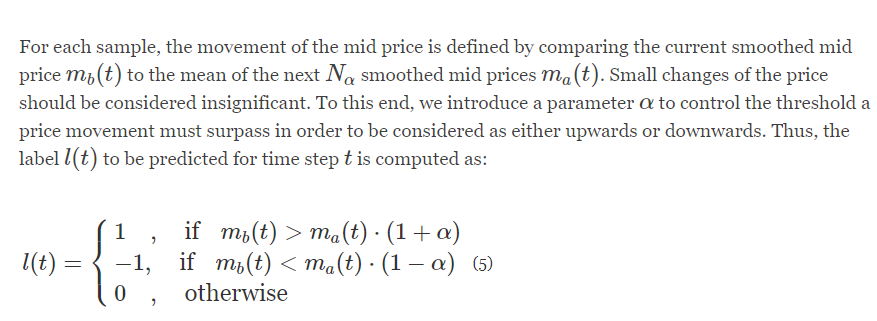
Of the values accompanying LOB data, **the tick price,** which is the price of the last executed trade, typically **varies wildly between the two sides of the margins**, introducing great amounts of noise to the prediction labels. The so-called market micro-structure **noise can be partially reduced by using mid-prices**, i.e. the mean of the best ask and best bid prices, **instead of transaction prices**.

Another advantage of using mid-prices instead of transaction prices is that **they are observable every time as long as there are orders on both bid and ask sides** while transaction prices are updated only at transactions.



N\_alpha = 1, 5, 10

Classification: up, down, no change



Alpha = 0.0001, 0.0002, 0.0003

Ex. mb(9) > ma(9) -> 上漲

lob\_ML\_3

Feature Engineering

**handcrafted features** and **fully automated feature extraction** methods

Since the problem under consideration in the present thesis focuses on financial

data (i.e., the mid-price prediction based on high-frequency data), relevant features/-

factors are related to financial concepts. More specifically, these features will be **LOB**

**features,** **technical indicators, quantitative analysis, econometrics, and fully automated**

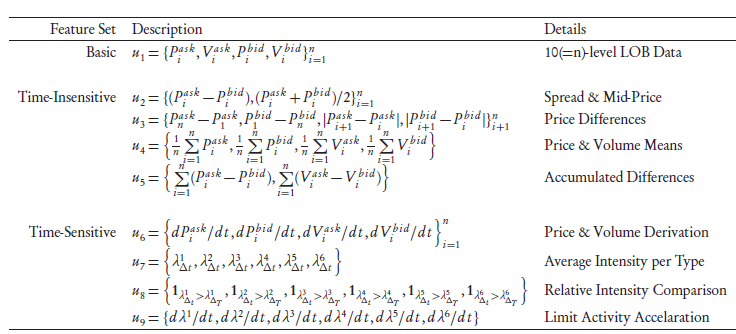
**financial processes.** The following list of features is by no means exhaustive,

demonstrating only a few exemplary cases.

LOB features

extraction is based on the prediction of mid-price. There are three main

categories of feature set whose description can be seen on below



Technical indicator

ADX, PSAR, LRL

Quantitative Analysis

Ex. Autocorrelation and Partial Correlation、Cointegration

Econometric

Ex. Vol, realized variance, realized kernel, realized pre-averaged variance

Fully Automated Feature Extraction

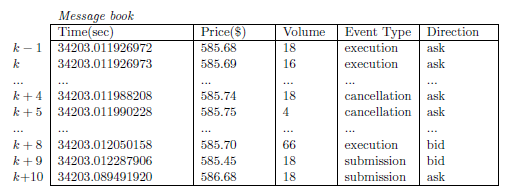
Ex. Autoencoders (AE)

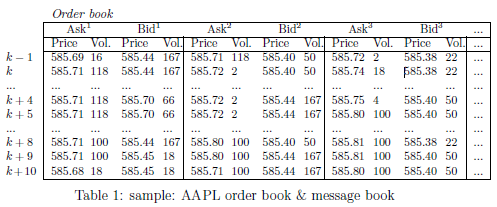
問題

Bid、ask price 有沒有1~10個levels的資料

Prediction多為mid price(t+1)的漲/跌/平 (classification)

Lob\_SVM





Event type: **limit order submission**, **limit order cancellation**, or **market order**

**execution**

It can also be observed from the message book that **multiple trading events** could arrive **within milliseconds** as demonstrated from Row k - 1 to k + 10, leading to **drastic fluctuation of prices**

**and volumesin** the order book. Although a variety of “metrics’’ have been designed to capture the price fluctuation, in this paper we select as metrics (a) the occurrence and direction of mid-price movement, and (b) the occurrence and direction of bid-ask spread crossing, described below.

These two metrics can work independently or together to provide guidance for

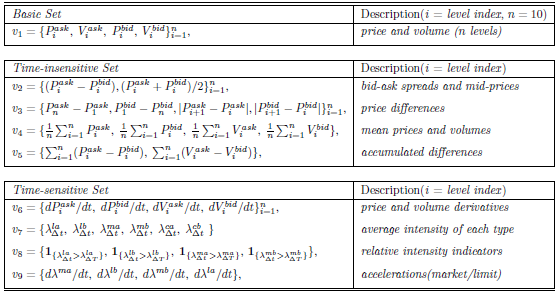
trading strategies depending on particular scenarios. Of course, the premise for the

arbitrage opportunities described above is that the directions of mid-price movement

and price spread crossing can be predicted accurately, which is the main task addressed

in this paper.

Feature Extraction and Training Data Preparation



7 days train 3 days test

3/9

10 levels bid ask price

**Basic set**

1. each bid ask price and their volume (OK)

**time insensitive**

1. bid-ask spread and mid price from 10 levels (OK)
2. bid, ask price diff (1-10、10 -1) and their abs value (OK)
3. bid, ask, their volume’s mean (OK)
4. accumulate diff (OK)

**time sensitive**

1. price and volume’s derivatives (per 5 sec) (OK)
2. average intensity: limit/ market/ cancellation ask/bid orders (per 5 sec) average arrival rate
3. relative intensity indicators: if 10sec’s rate > 900sec’s rate -> 1 else 0 (limit/ market)
4. v7’s derivative (per 5 sec) (limit/ market)

**需要的data**: 前10筆買賣價、event type(submission, execution, cancellation)、個股期貨資料

問題: rolling window(固定長度or累積?) roll幾筆? Test set 預測一筆就好? Response的表達方式

Tensor-based regression

Convolutional Neural Network (CNN)-based regression

Recurrent Neural Network (RNN)-based regression

結果: random forest ACC大概70%

3/10

Forecasting vix futures

LSTM RNN: 時間給予權重，愈遠的權重愈小 (long-short-term-memory RNN)

Contango: 遠期交割價 > 即期交割價

Backwardation: 遠期交割價 < 即期交割價

Input variables

1. term structure spreads: 不同到期日之vix future’s spread
2. vol surface skewness: The skew is the difference in implied volatility between the two strikes at a particular maturity.
3. Vix spot index & MA signals

(vix spot index have 14, 50, 100days)

If index cross upper MA -> +1, cross lower ma -> -1, others -> 0

1. VVIX(Vix of Vix): take its close price and its high-low intraday spread

3/11

防火牆擋住(http error 0000)解決方法

conda config --set ssl\_verify no

<https://towardsdatascience.com/installing-keras-tensorflow-using-anaconda-for-machine-learning-44ab28ff39cb>

conda update --all

cudatoolkit ver: 10.1.243

解決cuda版本問題

<https://blog.csdn.net/carolynlmk/article/details/86630865>

LSTM

<https://medium.com/@daniel820710/%E5%88%A9%E7%94%A8keras%E5%BB%BA%E6%A7%8Blstm%E6%A8%A1%E5%9E%8B-%E4%BB%A5stock-prediction-%E7%82%BA%E4%BE%8B-1-67456e0a0b>

看test

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

best one!

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

LSTM variable selection

<https://stats.stackexchange.com/questions/191855/variable-importance-in-rnn-or-lstm>

3/16

LSTM 整理

<https://zhuanlan.zhihu.com/p/31783805>

高頻交易 LSTM

<https://github.com/blockchain99/stock-predict-by-RNN-LSTM>

lstm early stopping

<https://blog.csdn.net/silent56_th/article/details/72845912>

deal with overfitting: early stop, drop out, regularization, batch normalization

drop out

<https://machinelearningmastery.com/how-to-reduce-overfitting-with-dropout-regularization-in-keras/>

response改成預測**報酬**

3/18

卡爾曼慮波 配對交易

Lstm normalized data will help the machine converge and get the features equal weighted

It will be beneficial to normalize your training data. Having different features with widely different scales fed to your model will cause the network to weight the features not equally. This can cause a falsely prioritisation of some features over the others in the representation.

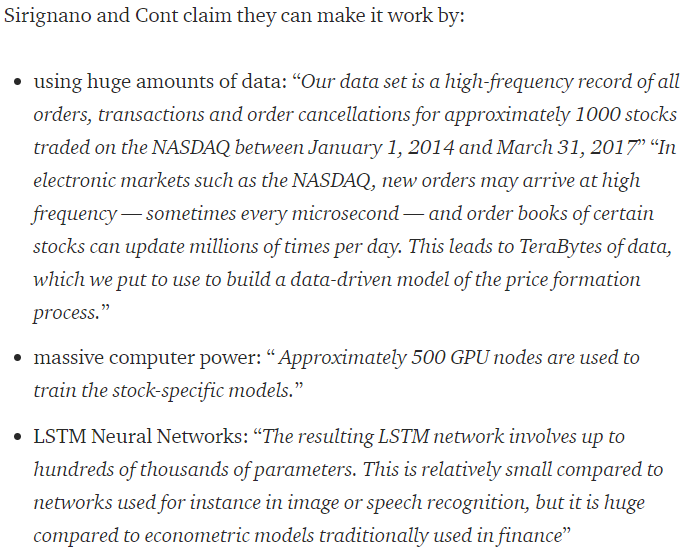
Despite that the whole discussion on data preprocessing is controversial either on when exactly it is necessary and how to correctly normalize the data for each given model and application domain there is a general consensus in Machine Learning that running a Mean subtraction as well as a general Normalization preprocessing step is helpful.

In the case of Mean subtraction, the mean of every individual feature is being subtracted from the data which can be interpreted as centering the data around the origin from a geometric point of view. This is true for every dimensionality.

Normalizing the data after the Mean subtraction step results in a normalization of the data dimensionality to approximately the same scale. Note that the different features will loose any prioritization over each other after this step as mentioned above. If you have good reasons to think that the different scales in your features bear important information that the network may need to truly understand the underlying patterns in your dataset, then a normalization will be harmful. A standard approach would be to scale the inputs to have mean of 0 and a variance of 1.

Better to predict in high freq dataset

1. with no noise, no large jump



Feature selection idea

Leave one out and compare the outcome

Upgrade colab’s RAM

<https://towardsdatascience.com/upgrade-your-memory-on-google-colab-for-free-1b8b18e8791d>

return -> 不變動的改成 後面往前補 -> 看看lstm會不會比較明顯

ex

mid: 126.75 126.75 126.75 126.75 127.25

return: ln(127.25/126.75) …………….