股票預測分析

判斷買多還是賣空、預判漲跌，使利潤最大化

Stock prediction and analysis

黃秉茂

Ping-Mao Huang

國立交通大學資訊學院資訊工程學系

Department of Computer Science

National Chiao-Tung University

中華民國 110 年 12 月

December, 2020

Abstract

AI 發展時至今日，在分群和辨識系統上已經做得很好了，人臉辨識、遊戲甚至是遠超人類，然而，在金融上的成就遠遠不及其他領域，也許是因為財金這塊領域的 domain knowledge 太深，導致模型在基礎設定上本來就考慮的不夠周全，因此我希望能運用機器學習看看 ML 是否能幫助股票分析。

Motivation

個人利益方面，當然是希望所學能幫自己賺錢。社會方面，因為 AI 在有些方面已經超越人類了，像是圍棋，而運用在財金方面是否能有顯著的提供協助?財金相較於其他領域是更貼近人們的生活的，因此可能對人類的生活產生巨變，而且投資要考量的變數實在很多，所以我很好奇運用 AI 技術的投資能否打敗人類。

AI 已經有發展到一定的程度了，然而運用 AI 的領域實在不算多，目前可能只有遊戲、視覺、跟自然語言處理有比較傑出的成果，AI 勢必還需要在某些能大大改變人們生活的領域發展，更能顯的 AI 會如何影響人類。如果 AI 加上財金將會大大地改變人們的生活，而且投資並不是只有考量數學，還是一門跟人文、心理相關的學問，而 AI 是否能連這些都考慮到是我所好奇的。

Introduction

在 AI 在其他領域有不凡的成就時，很多資料分析師都有個疑問，像是財金那麼複雜且和人們息息相關的領域，ML 有辦法去輔助人們做出決策嗎?以下是很多國外學者在股票分析上的成果，而我想看看當分析的資料不是全世界，而是僅僅是台灣時，會發生什麼事情?畢竟一般的台灣人通常只在乎國內的情形，很少對國外的金融商品下手。如果下列的成果都那麼好，那台灣的資料也會一樣好嗎?所以我會透過技術面去預測及分析股價。

Related Work

- "Global stock market investment strategies based on financial network indicators using machine learning techniques. Lee, Tae

Kyun, et al. "Global stock market investment strategies based on financial network indicators using machine learning techniques." Expert Systems with Applications 117 (2019): 228-242.

- "Supporting Investment Management Processes with Machine Learning Techniques." Groth, Sven S., and Jan Muntermann. "Supporting Investment Management Processes with Machine Learning Techniques." Wirtschaftsinformatik (2). 2009.
- "A machine learning model for stock market prediction." Hegazy, Osman, Omar S. Soliman, and Mustafa Abdul Salam. "A machine learning model for stock market prediction." arXiv preprint arXiv:1402.7351 (2014).
- "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques." Patel, Jigar, et al. "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques." Expert systems with applications 42.1 (2015): 259-268.
- 希望能用台灣的股票資料，做出和上述差不多好的結果

Methodology

Problem statement - Technical analysis

1. 起始金額為現金 10,000 元，希望最後能賺最多錢
2. 每天都要判斷買多還是賣多這固定的十檔股票(0050、0056、鴻海、台積電、聯發科、大立光、富邦金、國泰金、玉山金、元大金)
3. 自變數會用先前的開盤價、最高價、最低價、收盤價、成交量、5 日均價、20 日均價、網路聲量(如果有學會爬蟲的話)等等，而變數就是這十檔分別該買還是賣
4. 假設想買一定買得到，想脫手一定賣得掉
5. 假設股票能以前一天的收盤價買進，融券也是以前一天的收盤價為代價
6. 假設手續費為交易金額的 0.1%
7. 先預測哪些會漲那些會跌之後，先將現金 10,000 元保留以備不時之需。在認為漲超過手續費的股票中，覺得投資報酬率最高的投入剩餘現有現金的 50%，第二高的再投入剩餘現有現金的 50%，以此類推，並將剩餘所有的現有現金投入投

資報酬率最低但還是認為能漲超過手續費的，然後隔天要將所有持有的股票都賣掉。認為會跌超過手續費的也是按照買股的方式，在認為跌超過手續費的股票中，覺得賣空的投資報酬率最高的就借價值約為現有現金的 50%的券，賣空的投資報酬率第二高的再借價值約為剩餘現有現金的 25%的券，以此類推，並在最後借與前一個同價值但還是認為能跌超過手續費的券，然後隔天也要將所有持有的股票都賣掉平倉。

8. 可以全部都不買也不賣

Baseline Method

- 將所有的變數拿去跑回歸，並預測報酬率
- 預測股價與目前股價做比較
- Rolling window 研究 365 天的資料決定第 366 天的買賣
- Testing data 也是 365 天

ML Method

- 利用 PCA 找出重要參數，並且減少運算量
- 跑 SVM 和 DNN，並預測報酬率
- 預測股價與目前股價做比較
- Rolling window
- 每一次都是研究 365 天的資料決定第 365 天的買賣
- Testing data 也是 365 天

超參數設定

PCA: n_components = 24

DNN: hidden layer 1: 10 neurons, selu

hidden layer 2: 10 neurons, relu

output layer: 1 neurons 預測報酬率

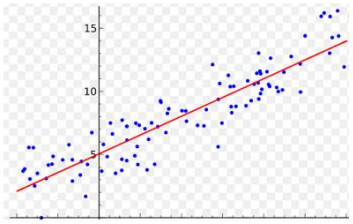
用 selu 和 relu 因為是線性關係

loss: mse optimizer: adam metrics: mae

DNN 基本上能做到 linear regression 作得到的事，所以效果應該會比較好

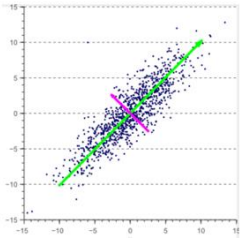
Linear regression

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^{\mathsf{T}} \boldsymbol{\beta} + \varepsilon_i$$



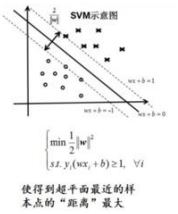
PCA

$$\mathbf{w}_k = \arg\max_{\|\mathbf{w}\|=1} E\left\{\left(\mathbf{w}^{\top}\hat{\mathbf{X}}_{k-1}\right)^2\right\}$$



SVM

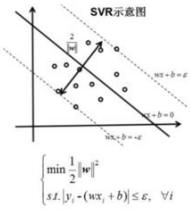
$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i$$
$$\text{subject to } y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i,$$
$$\zeta_i \geq 0, i = 1, \ldots, n$$



使得到超平面最近的样本点的“距离”最大

SVR

$$\min_{w,b,\zeta,\zeta^*} \frac{1}{2} w^T w + C \sum_{i=1}^{n} (\zeta_i + \zeta_i^*)$$
$$\text{subject to } y_i - w^T \phi(x_i) - b \leq \varepsilon + \zeta_i,$$
$$w^T \phi(x_i) + b - y_i \leq \varepsilon + \zeta_i^*,$$
$$\zeta_i, \zeta_i^* \geq 0, i = 1, \ldots, n$$



使得到超平面最远的样本点的“距离”最小

DNN

$$\Delta w_{ij}(t+1) = \Delta w_{ij}(t) + \eta \frac{\partial C}{\partial w_{ij}}$$

mse

$$\frac{1}{n}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2$$

mae

$$MAE = \frac{\sum_{i=1}^{n} | y_i - \hat{y}_i |}{n}$$

adam

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\frac{\partial L_t}{\partial W_t}$$
$$v_t = \beta_1 v_{t-1} + (1-\beta_2)(\frac{\partial L_t}{\partial W_t})^2$$
$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$
$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$
$$W \leftarrow W - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

Data

data source: TEJ, yahoo finance, apple stock

會先將.csv 下載下來後，再使用 pandas, numpy 等進行 EDA

stock:

- 0050 Yuanta Taiwan Top5: 0050 元大台灣 50
- 0056 PTD: 0056 元大高股息
- 2317 Hon Hai Precision: 2317 鴻海
- 2330 TSMC: 2330 台積電
- 2454 MediaTek: 2454 聯發科
- 3008 Largan: 3008 大立光
- 2881 Fubon FHC: 2881 富邦金
- 2882 Cathay Holdings: 2882 國泰金
- 2884 E.S.F.H: 2884 玉山金
- 2885 Yuanta Group: 2885 元大金

attribute:

- CO_ID: 公司代碼
- Date: 年月日
- Open(NTD): 開盤價(元)
- High(NTD): 最高價(元)
- Low(NTD): 最低價(元)
- Close(NTD): 收盤價(元)
- Volume(1000S): 成交量(千股)
- Amount(NTD1000): 成交值(千元)
- AVG CLOSE: 當日均價(元)
- AVG CLOSE 5D: 5 日均價(元)
- AVG CLOSE 10D: 10 日均價(元)
- AVG CLOSE 20D: 20 日均價(元)
- AVG Vol 5D: 5 日均量
- AVG Vol 10D: 10 日均量
- AVG Vol 20D: 20 日均量
- ROI%: 報酬率%
- Shares(1000S): 流通在外股數(千股)
- Market Cap.(NTD MN): 市值(百萬元)
- P/E-TEJ: 本益比-TEJ
- P/B-TEJ: 股價淨值比-TEJ
- Dividend_Yield%: 股利殖利率

- Cash_Dividend%: 現金股利率
- Price_Change(NTD): 股價漲跌(元)
- High minus Low %: 高低價差%
- Market: 上市別
- Capital: 資本
- No.of Employee: 員工人數

EDA(Exploratory Data Analysis)

data clean-ups (data preprocess for dropping column with lots of NaN and replacing NaN with mean)

categorical attribute and continuous attribute

Statistical Computing and Data Visualization

Data type transformation and transform data format and shape so your model can process them.

drop some useless attribute

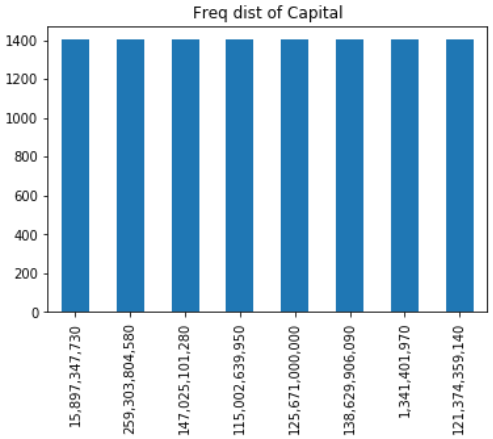
PCA

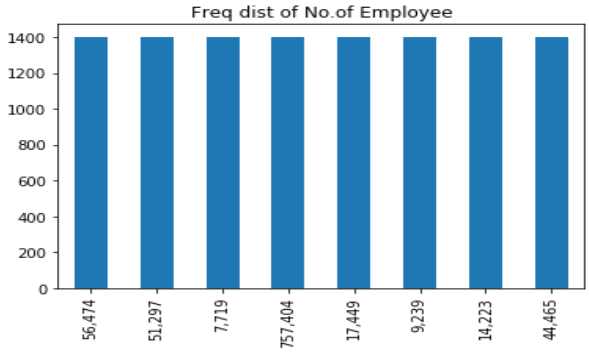
create rolling window (1 day → 365 days)

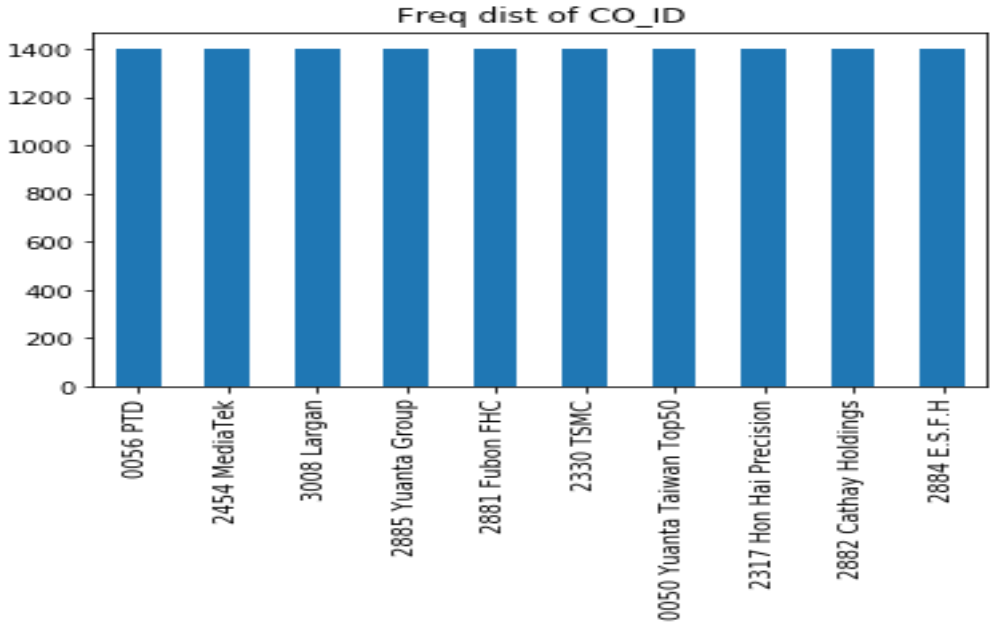
y: 'ROI'

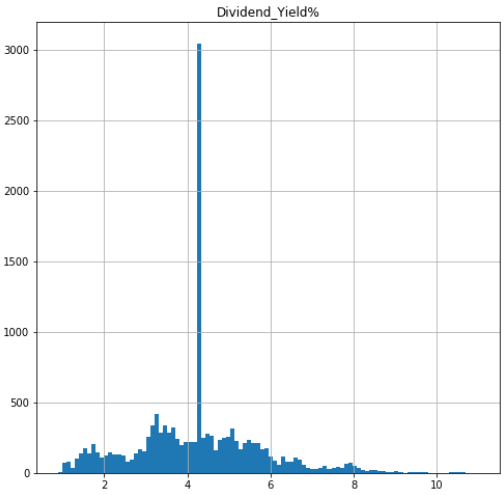
Shuffle the data

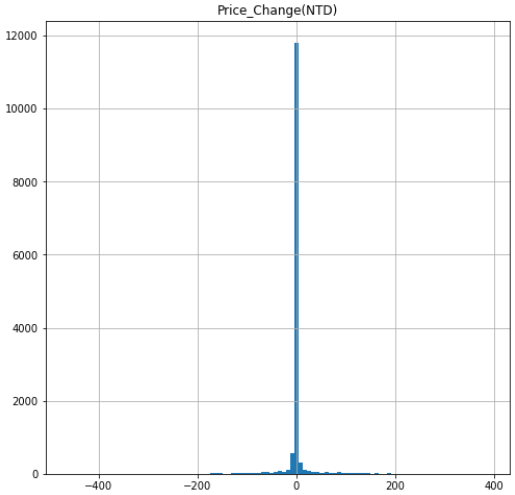
	ROI%	P/E-TEJ	P/B-TEJ	Dividend_Yield%	Cash_Dividend%	Price_Change(NTD)	High minus Low %
count	14020.000000	14020.000000	14020.000000	14020.000000	14020.000000	14020.000000	14020.000000
mean	0.046360	15.044964	2.296988	4.227955	3.663234	0.139675	1.391597
std	1.617399	6.357659	1.845026	1.513258	1.391331	32.966458	1.234087
min	-10.000000	5.530000	0.590000	0.890000	0.000000	-465.000000	0.000000
25%	-0.662300	10.760000	1.110000	3.320000	2.654900	-0.350000	0.659900
50%	0.000000	14.820000	1.660000	4.227955	3.590000	0.000000	1.183400
75%	0.732825	16.240000	2.296988	5.020000	4.540000	0.400000	1.800150
max	10.000000	47.550000	12.200000	11.000000	11.000000	390.000000	12.797600

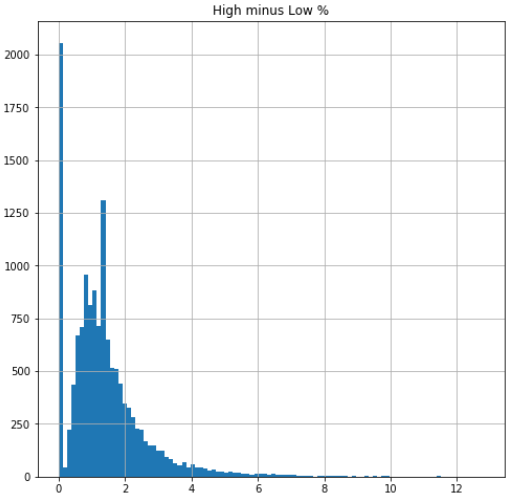


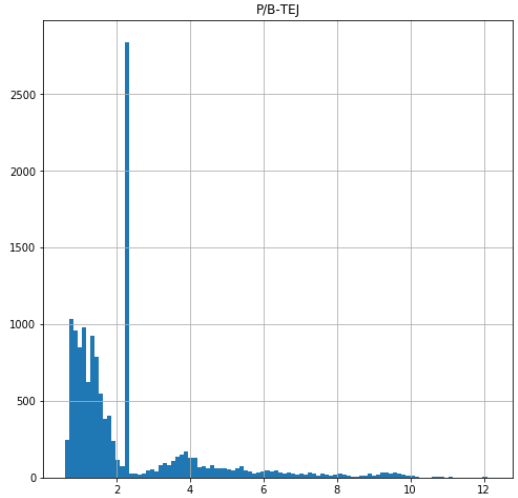


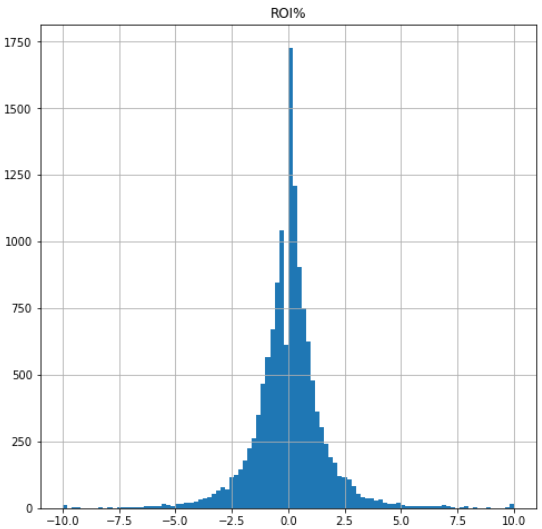


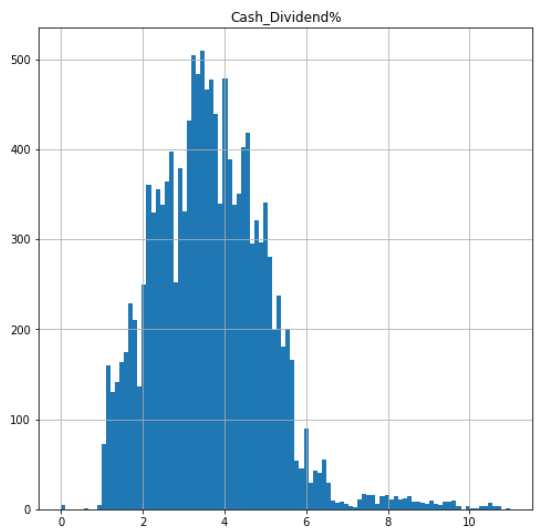

Dividend_Yield%
3000
2500
2000
1500
1000
500
0
2
4
6
8
10

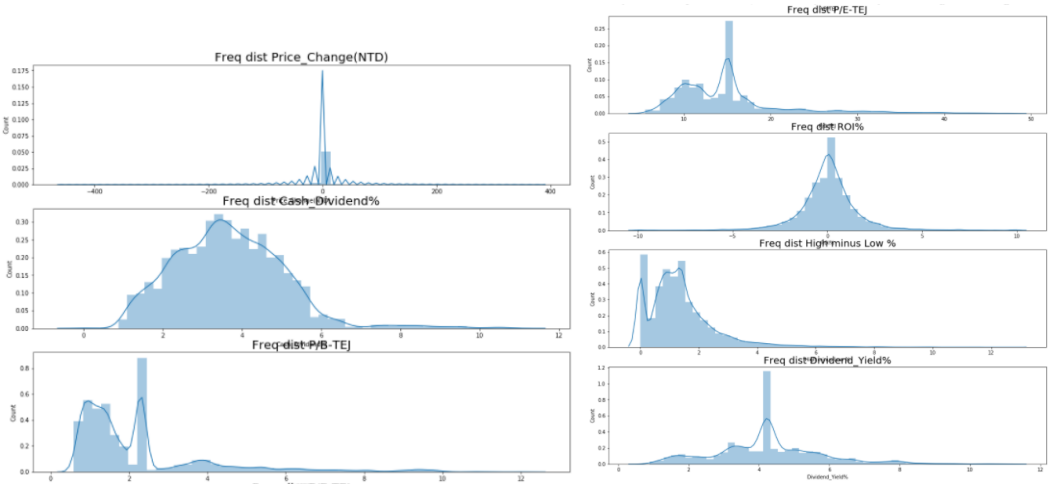

Price_Change(NTD)
12000
10000
8000
6000
4000
2000
0
−400
−200
0
200
400

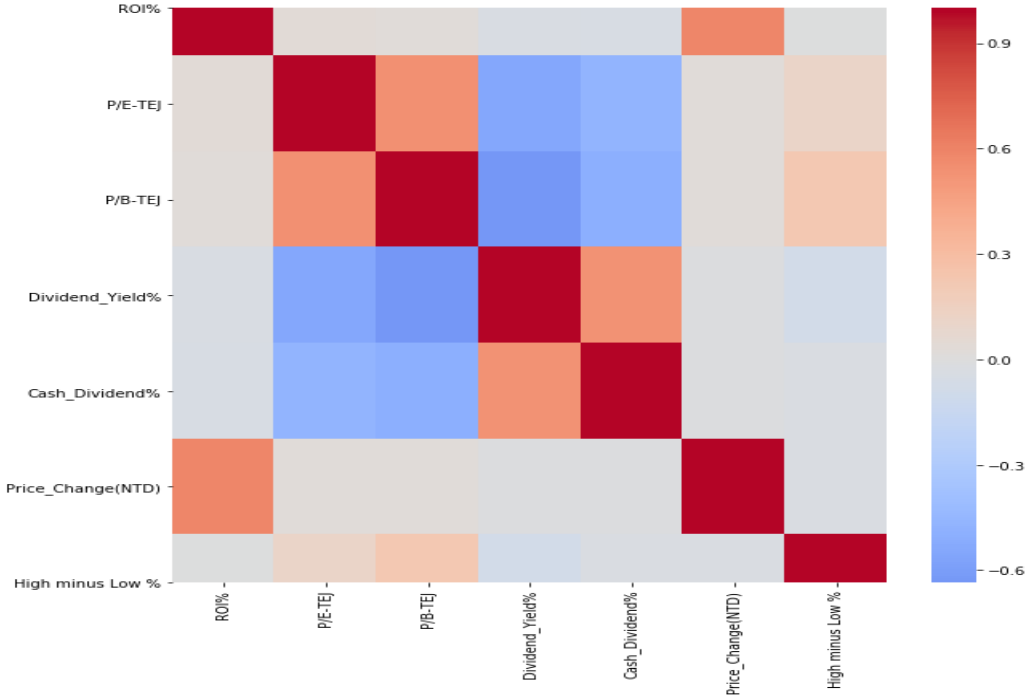

High minus Low %
2000
1750
1500
1250
1000
750
500
250
0
0
2
4
6
8
10
12


P/B-TEJ
2500
2000
1500
1000
500
0
2
4
6
8
10
12


ROI%
1750
1500
1250
1000
750
500
250
0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


Cash_Dividend%
500
400
300
200
100
0
0
2
4
6
8
10


Freq dist Price_Change(NTD)
Freq dist Cash_Dividend%
Freq dist P/B-TEJ
Freq dist P/E-TEJ
Freq dist ROI%
Freq dist High minus Low %
Freq dist Dividend_Yield%


ROI%
P/E-TEJ
P/B-TEJ
Dividend_Yield%
Cash_Dividend%
Price_Change(NTD)
High minus Low %
0.9
0.6
0.3
0.0
−0.3
−0.6

Results

Observe day → rolling window test 30 days

	Linear regression	SVM	DNN
observe 30 days	11025.41257057684	11424.564134181688	11720.512937545913
observe 90 days	11139.629688138475	11115.14139858359	11237.980129592774
observe 180 days	11005.666149997362	11115.14139858359	11466.011329181278

test 365 days

	Linear regression	SVM	DNN
observe 365 days	47304.72179350837	48430.53678658222	70493.21038143926

All bigger than initial amount ($10,000)

只用 30 天都能從 10000 賺到 11000 以上，可以看出三種模型都是有幫助的，而測試一年的話 DNN 更是能賺到 7 萬，考慮這麼久基本上不會只是多頭或空頭市場。結果也能看出 ML 和 DL 的確是有用的技術，能幫助我們賺更多錢，因為基本上賺的都比儲蓄高，可以認為對預測買賣是有一定的效力的。

Conclusion

隨著 rolling window 變大，DNN 效果更顯著，而因為完全沒有人為因素，且流通性假設為很高，所以才能賺到這個數字，也代表了少了很多人為影響決策的因素，散戶也能透過智慧理財賺錢，而這是在相對保守的投資策略下賺錢，也許更激進點的能夠賺一大筆錢。

Future Use

希望能透過更多 ML 的方法，甚至是 DL 和 RL 的方法，讓成果變得更好，未來也會考慮考量更多變數並篩掉不太有解釋效用的變數，也希望未來能不僅僅從技術面分析，也考量到基本面分析，這樣不僅能做的更全面，也更能分辨哪些因素才是重要因子，又或是透過文字探勘蒐集其他有效的資訊，也期許未來能考量更多市場機制，例如流動性等等。