巨量資料與金融科技實務之應用—預測股票因子

資科二 B 李嘉芸

指導教授：鄭宏文老師

目錄

主題

利用動能因子結合 AI 機器學習去預測未來動能因子並建立市場投資組合，與大盤進行比較。

動能因子(5 種)：1 個月,6 個月,12 個月,36 個月,60 個月

AI 機器學習(3 種) ：線性迴歸,隨機森林,類神經網路

動機

以臺灣 900 多家上市上櫃公司的股票市價作為資料，利用因子策略去計算歷史 IC 值（因子與未來股票報酬的相關係數）。使用三種 AI 機器學習（線性迴歸、隨機森林及類神經網路）來預測每個風格因子未來一期的 IC 值，再根據預測的 IC 值對風格因子賦予之權重，利用買高賣低或買低賣高的策略建構市場投資組合與大盤做比較，希望藉由 AI 機器學習方法來提高勝率及穩定性。

方法

(假設持有期間為 1 個月)

1. 求出 1 個月、6 個月、12 個月、36 個月、60 個月動能因子的 IC 值。

2.將歷史 IC 值丟入預測模型（線性迴歸、隨機森林及類神經網路）預測下一個月 IC 值，X 為動能因子值，Y 為 IC 值，會得到 5 個不同動能因子的 IC 值。

3.從 5 個 IC 值中取絕對值大的值作為對下個月報酬較敏感的因子。

4. 若 IC 值為正，投資組合策略則是買高等分賣低等分。若 IC 值為負，投資組合策略則是買低等分賣高等分。

5. 假設選到的是 1m 動能因子，排序所有公司 1m 動能因子值的大小，分成 10 等分。假設投資策略為買高賣低，買入最高等分的所有公司股票，賣掉最低等分所有公司股票，計算持有一個月的投資組合報酬率。

6. 再去抓台灣指數的月收盤價並計算報酬率，與自己的報酬率做比較，看是否有打敗大盤。

其他的分析：

1. 改變排序等份：可以用多種等分做比較，找出報酬率較好的投資組合。

2. 可以改變預測的時間點：增加樣本點，更精確的預測結果。

3. 改變不同的 AI 機器學習程式：可以比較預測結果找出更精確之機器學習程式。

資料來源

此股價數據來自台灣經濟新報(TEJ)資料庫，資料期間從 1999 年 12 月至 2019 年 1 月，共 18 年。

資料處理

主要會用到的資料為動能因子、月報酬率、IC 值

- 動能因子：本期股票收盤價除以前 n 期股票收盤價取對數

  2000 年 1 月的 1 個月動能=LOG(2000 年 1 月價格/1999 年 12 月價格)

- 月報酬率：下期股票收盤價除以當期股票收盤價取對數

  2000 年 1 月的未來 1 個月報酬=LOG(2000 年 2 月價格/2000 年 1 月價格)

- IC 值：動能因子和月報酬率之相關係數

(五個動能因子和未來 1 個月報酬皆用相同方式進行資料洗清，以下由一個月的動能因子為例。)

1. 首先清除工作表內錯誤的資料，把資料中的 DIV/0!和 NUM!清除。資料是由公式組成，因此將整張工作表複製後，在新工作表以值的方式貼上並命名為『一個月動能_缺值』。去尋找與選取的地方，以「 」分別取代全部 DIV/0!和 NUM!。
2. 用函數=IF(OR('1 個月動能' ="",未來 1 個月報酬=""),"",'1 個月動能')補上工作表中的缺值，如果『1 個月動能』這張表儲存格是空的，或是『下個月報酬』的儲存格是空的，那儲存格就會變成空白，如果兩張表都有資料話則會顯示該張表原先的數值。複製後開一個新的工作表命名為『一個月動能_缺值』以值的方式貼上，再使用函數=IF('1 個月動能_缺值' ="",MEDIAN('1 個月動能_缺值'),'1 個月動能_缺值')，把能是空值的資料轉換成該月之中位數，即完成資料清洗。
3. 新工作表「1 個月動能_補值_rank」使用函數=RANK('1 個月動能_補值','1 個月動能_補值',1)將資料以遞增的方式排序，未來 1 個月報酬也使用同樣方式排序。
4. 用函數=CORREL('1 個月動能_補值_rank',’未來 1 個月報酬_補值_rank’)計算兩個資料群的相關係數，即為 IC 值。

程式撰寫

(以下由 1 個月的預測模型為例，相同步驟即可預測不同變因之模型)

```
import os
import pandas as pd
import numpy as np
import tensorflow as tf
from sklearn import ensemble
from sklearn.ensemble import RandomForestRegressor,GradientBoostingRegressor
from sklearn.linear_model import LinearRegression
from keras.models import Sequential
from keras.layers import Dense,Activation
from keras.models import load_model
from tensorflow.keras import optimizers
import tensorflow as tf
```

1. 導入所需之套件

   os: 檔案管理/pandas: 資料處理與分析/numpy: 陣列處理、提供大量數學函式庫/tensorflow:深度學習/sklearn:機器學習/keras: 神經網路

   利用 as 可自訂導入的套件名稱，使用時即可簡化程式碼方便閱讀。

```
path=r"D:/SCU"
#改變當前目錄
os.chdir(path)
```

2. 指定路徑

   需要讀入的資料要放在此資料夾中，程式碼和寫出的資料也會存在此。

```
#讀取1m值資料並清空NaN
df_x_1m = pd.read_excel('10173207_mom1.xlsx',sheet_name='1個月動能_補值',header=1,index_col=1)
df_x_1m = df_x_1m.iloc[:,2:].dropna()
#讀取1m ic值資料並清空NaN
df_y_1m = pd.read_excel('10173207_mom1.xlsx',sheet_name='1個月動能_IC值',header=1,index_col=0)
df_y_1m = df_y_1m.iloc[:,:-1].dropna() #axis=1
```

3. 讀取資料

在 pd.read.excel()的()中分別放 (檔名、所需之工作表、上標設定、左標設定)。用下圖做為範例，由於 python 從 0 開始計算，若要設定上標為時間，則 header=1，而左標為公司名稱即 index_col=1。

	A	B	C	D	E	F	G	H	I	J
1		0		2	3	4	5	6	7	8
2	0	證券代碼	ID	2000/01/31	2000/02/29	2000/03/31	2000/04/29	2000/05/31	2000/06/30	2000/07/31
3	1	1101 台泥	1101	0.0726538	0.0512785	-0.0991832	-0.1373541	0.0476642	-0.0582144	0.0070619
4	2	1102 亞泥	1102	0.0376574	0.0373073	-0.0687158	-0.09691	0.01524	-0.064458	0.0530784
5	3	1103 嘉泥	1103	0.0634686	0.011281	-0.0408363	-0.0683082	0.0163904	-0.0211893	0.0522235
6	4	1104 環泥	1104	0.0753548	-0.0236947	-0.0229177	-0.0474246	-0.0047989	-0.0820862	0.0199383
7	5	1108 幸福	1108	0.0663259	0.0055561	-0.0632818	-0.0354723	-0.0188853	-0.0457575	0.0053288
8	6	1109 信大	1109	0.0753703	-0.022664	-0.0015538	-0.0582144	0.0309115	-0.0472334	0.0036805
9	7	1110 東泥	1110	0.0530784	-0.0236947	-0.0207836	-0.0766341	0.0100419	-0.044381	0.0135305
10	8	1201 味全	1201	0.2365187	0.0030801	0.0106116	-0.0767917	-0.0511525	-0.0767752	0.0071392
11	9	1203 味王	1203	0.0937968	0.0445059	0.0701579	-0.0401946	-0.1171134	0.0955291	-0.0591791

df_x_alpha.iloc[:,2:].dropna()清除不需要的資料(0、ID 兩欄)，用 iloc 選取 dataframe 中的資料，dropna()則是用來清除 Nan 資料。[]中逗號左邊是選取列，右邊則是選取哪幾欄。可以選擇選取範圍，舉例來說: 「:」的意思為選取全部，「3:5」的意思為選取第 3~第 5 行，「2:」的意思為選取前 2 行，「:2」的意思為選取最後 2 行。

證券代碼	000-01-31 00:00:0	000-02-29 00:00:0	000-03-31 00:00:0	000-04-29 00:00:0	000-05-31 00:00:0	000-06-30 0
1101 台泥	0.0726538	0.0512785	-0.0991832	-0.137354	0.0476642	-0.05821
1102 亞泥	0.0376574	0.0373073	-0.0687158	-0.09691	0.01524	-0.06445
1103 嘉泥	0.0634686	0.011281	-0.0408363	-0.0683082	0.0163904	-0.02118
1104 環泥	0.0753548	-0.0236947	-0.0229177	-0.0474246	-0.00479888	-0.08208
1108 幸福	0.0663259	0.00555608	-0.0632818	-0.0354723	-0.0188853	-0.04575
1109 信大	0.0753703	-0.022664	-0.00155383	-0.0582144	0.0309115	-0.04723
1110 東泥	0.0530784	-0.0236947	-0.0207836	-0.0766341	0.0100419	-0.04438
1201 味全	0.236519	0.00308012	0.0106116	-0.0767917	-0.0511525	-0.07677
1203 味王	0.0937968	0.0445059	0.0701579	-0.0401946	-0.117113	0.095529

讀取後的 df_x_1m

```
#轉置
df_x_1m=df_x_1m.T
df_y_1m=df_y_1m.T
```

4. 轉置

將時間和公司名稱的位置互相調換，方便後續之資料整理

Index	1101 台泥	1102 亞泥	1103 嘉泥	1104 環泥	1108 幸福	110
2000-01-31 00:00:00	0.0726538	0.0376574	0.0634686	0.0753548	0.0663259	0.075
2000-02-29 00:00:00	0.0512785	0.0373073	0.011281	-0.0236947	0.00555608	-0.02
2000-03-31 00:00:00	-0.0991832	-0.0687158	-0.0408363	-0.0229177	-0.0632818	-0.001
2000-04-29 00:00:00	-0.137354	-0.09691	-0.0683082	-0.0474246	-0.0354723	-0.058
2000-05-31 00:00:00	0.0476642	0.01524	0.0163904	-0.00479888	-0.0188853	0.030
2000-06-30 00:00:00	-0.0582144	-0.064458	-0.0211893	-0.0820862	-0.0457575	-0.047
2000-07-31 00:00:00	0.00706185	0.0530784	0.0522235	0.0199383	0.00532883	0.003
2000-08-31 00:00:00	-0.102196	-0.126222	-0.146128	-0.110339	-0.0273231	-0.08
2000-09-30 00:00:00	-0.0669468	-0.0848719	-0.0183532	-0.065752	-0.138303	-0.044
2000-10-31 00:00:00						

轉置後的 df_x_1m

```
#設定線性、隨機森林訓練集
#1m值(2000/1-2018/11)
X_1m =df_x_1m.iloc[:-2,:].values
Y_1m =df_y_1m.iloc[1:-1,:].values
```

5. 設定線性回歸和隨機森林訓練集

取全部公司 2000/1~2018/10 的動能，由於動能的資料為 2000/1~2018/12，用「:2」選取行列的範圍、用「:」選取所有欄位的範圍，IC 值也適用相同方法取 2000/2~2018/11 作為訓練集。

```
#設定NN1~NN3訓練集
#1M值(2000/1~2011/12)
X_1m_train = df_x_1m.iloc[:165,:].values
#1m ic值 (2000/1~2011/12)
Y_1m_train = df_y_1m.iloc[1:166,:].values

#設定NN1~NN3驗證集
#1M值(2012/1~2018/11)
X_1m_validation = df_x_1m.iloc[165:-2,:].values
#1m ic值 (2012/1~2018/11)
Y_1m_validation = df_y_1m.iloc[166:-1,:].values
```

6. 設定 NN1 訓練集和驗證集

用上述方法取 2000/1~2013/9 的動能和 2000/2~2013/10 的 IC 值作為訓練集，取 2013/10~2018/10 的動能和 2013/11~2018/11 的 IC 值作為驗證集。

```
#設定測試集
#2018/12的1m
test_data_1m = df_x_1m.iloc[-2:-1,:].values
```

7. 設定測試集

取 2018/12 的動能作為測試集

```
#建立模型
#回歸建模
reg=LinearRegression()
#隨機森林建模
regr=RandomForestRegressor(max_depth=2,random_state=42,n_estimators=100)
#NN1建模
model_1m_NN1 = Sequential()
model_1m_NN1.add(Dense(units=32,activation='relu',input_dim=945))
model_1m_NN1.add(Dense(units=1))
adam = tf.keras.optimizers.Adam(learning_rate=0.001)
my_seed = 42
tf.random.set_seed(my_seed) #種子讓結果相同
model_1m_NN1.compile(optimizer=adam,loss='mae')
```

8. 建立線性迴歸、隨機森林、類神經網路模型

```
#丟入數據訓練
reg.fit(X_1m,Y_1m)
regr.fit(X_1m,Y_1m)
model_1m_NN1.fit(X_1m_train,Y_1m_train,validation_data=(X_1m_validation,Y_1m_validation),
                 batch_size=32,epochs=10)
```

9. 丟入數據訓練

```
#預測(1m)
print('1m')
print("==========")
print('線性回歸',reg.predict(test_data_1m))
print('隨機森林',regr.predict(test_data_1m))
print('NN1',model_1m_NN1.predict(test_data_1m))
```

10.輸出結果

```
1m
==========
線性回歸 [[0.05840322]]
隨機森林 [0.00672566]
1/1 [==============================] - 0s 46ms/step
NN1 [[0.06386159]]
```

1 個月動能預測出的 2018/12 IC 值

成果展示

(以線性回歸為例，選出六個月的資料做投資組合策略)

1. 從模型中選絕對值大的那組並利用該月資料做投資組合策略。若選出之 IC 值為正，投資組合策略是買高等分賣低等分，反之，則為買低等分賣高等分。

	線性迴歸
1m	0.06386159
6m	-0.60580342
12m	-0.11229699
36m	-0.28755439
60m	0.05096589

2. 預測之 IC 值是負數，策略則為買低等分賣高等分。將動能因子值由低至高排序分成數等分，第 1 等分的所有公司為最低等分的投資組合，最後一等分的所有公司即為最高等分的投資組合。

公司	6個月動能	rank	下個月報酬
1529 樂士	0.744727495	1	0.039239409
1515 力山	0.251161669	2	0.021189299
6552 易華電	0.245735849	3	-0.001660784
3701 大眾控	0.241914835	4	-0.006681585
6230 超眾	0.197871033	5	-0.003058424
910708 恒大健-DR	0.189455221	6	0.008639655
2702 華園	0.182178505	7	0.008446836
4927 泰鼎-KY	0.179561847	8	-0.009433668
5259 清惠	0.178246571	9	-0.024823584

3. 假設投資在每間公司的比重是相同的，於 2018 年 12 月買這個投資組合，持有一個月算報酬率，上圖的下個月報酬率即為 2019 年 1 月的報酬率。
4. 此線性迴歸策略為買低賣高，預測之報酬率即為最低等分減最高等分。

線性迴歸	
預測2018/12 IC	-0.60580342
策略	買低賣高
等分	10
高	0.012393804
低	0.010983374
預測	-0.00141043

5. 下載台灣指數的月價格，計算 2019 年 1 月的 log 報酬率跟我們的投資組合報酬率做比較，是否有打贏大盤。

等分	10	20	50
高	0.012393804	0.008926055	0.0043971
低	0.010983374	0.00908086	0.008193909
預測	-0.00141043	0.000154804	0.003796808
大盤漲跌	0.009050858	0.009050858	0.009050858
輸贏	大盤贏	大盤贏	大盤贏
每份的數量	95	47	19

6. 做完以上步驟既是用線性迴歸模型預測出 2018 年 12 月 IC 值，再用選到的動能因子排序分等分來建構投資組合跟大盤做比較看是否有打贏大盤。若分成不同等分，結果也會有所差異。

成果分析

線性迴歸取六個月的資料；隨機森林取 60 個月的資料；類神經網路 NN1 取 12 個月的資料

	線性迴歸	隨機森林	類神經網路NN1
1m	0.06386159	0.00672566	-0.04957483
6m	-0.60580342	0.01727917	0.1381634
12m	-0.11229699	0.03440186	0.20796822
36m	-0.28755439	-0.00653416	-0.04796672
60m	0.05096589	-0.03713492	-0.20282932

線性迴歸

線性迴歸						
預測2018/12 IC	-0.60580342		2018/12/30台股指數	9727.41		
策略	買低賣高		2019/1/30台股指數	9932.26		
等分	10	20	50	100	200	300
高	0.012393804	0.008926055	0.0043971	0.003539684	0.009805583	0.019589
低	0.010983374	0.00908086	0.008193909	0.010304443	0.018111089	0.003481
預測	-0.00141043	0.000154804	0.003796808	0.006764759	0.008305506	-0.01611
大盤漲跌	0.009050858	0.009050858	0.009050858	0.009050858	0.009050858	0.009051
輸贏	大盤贏	大盤贏	大盤贏	大盤贏	大盤贏	大盤贏
每份的數量	95	47	19	9	5	3

隨機森林

隨機森林						
預測2018/12 IC	-0.03713492		2018/12/30台股指數	9727.41		
策略	買低賣高		2019/1/30台股指數	9932.26		
等分	10	20	50	100	200	300
高	0.01365106	0.021447703	0.033119243	0.013643046	0.012464712	0.010623551
低	0.011956211	0.01613882	0.01994015	0.033557789	0.028459193	0.04049561
我們預測	-0.001694849	-0.005308884	-0.013179093	0.019914743	0.01599448	0.029872058
大盤漲跌	0.009050858	0.009050858	0.009050858	0.009050858	0.009050858	0.009050858
輸贏	大盤贏	大盤贏	大盤贏	我們贏	我們贏	我們贏
每份的數量	95	47	19	9	5	3

類神經網路 NN1

類神經網路NN1						
預測2018/12 IC	0.20796822		2018/12/30台股指數	9727.41		
策略	買高賣低		2019/1/30台股指數	9932.26		
等分	10	20	50	100	200	300
高	0.015087973	0.01602657	0.026567221	0.042048544	0.077156138	0.120105692
低	0.012540561	0.01079721	0.008804463	0.013449023	0.01629418	0.006497229
我們預測	0.002547411	0.00522936	0.017762758	0.028599521	0.060861958	-0.113608463
大盤漲跌	0.009050858	0.009050858	0.009050858	0.009050858	0.009050858	0.009050858
輸贏	大盤贏	大盤贏	我們贏	我們贏	我們贏	我們贏
每份的數量	95	47	19	9	5	3

結論

用不同的模型會有不同的結果，每等分的數量越小，則能將結果分析得夠詳細，就會發現我們贏的機率會比大盤高。

心得

不同於其他的課程，在這堂課中需要實際操作，接觸到完全不同的領域。一開始花了很多時間在理解新接觸的名詞、熟悉作業環境，老師仔細的解釋並在每堂課的抽問和助教一步一步帶領，我才漸漸上手。就像在做一個完整的專案，從抓取到分析資料並從中找到其價值，課堂上教的技巧也可以應用在國外的股市做分析，甚至可以再延伸用更多模組預測，是一個很不錯的經驗。