台灣 ETF 價格預測競賽
設計文件

隊伍: [Yi]
成員：[張家豪],[江哲宇]

摘要

1.資料處理

將原始資料都轉成 row 為日期(無重複)，col 為所有特徵。針對 Y 的部分做一階差分，相當於將收盤價取代為每日的漲幅。X 的部分若出現遺失值，直接把這個特徵移除。由於要做 5 期預測，將資料集 X 往下移 5 期當做預測要用的資料，Y 的資料會多出前 4 筆直接捨棄。

示意圖如下:

丟棄資料	y 1	
	y 2	
	…	
	y 4	
訓練資料	y 5	x 1
	…	…
	y n-1	x n-5
預測用資料		x n-4
		x n-3
		x n-2
		x n-1
		x n

2.特徵的選取上沒有特別偏好，使用了調整後股價的所有資訊(開盤、收盤、最高值、最低值、交易張數)，依據不同的模型會有不同特徵選擇的標準。唯一增加了每天的星期天數，想創造出時間性對反應變數的影響。當然有很多增加變數的方法，主要的重心放在模型的參數調整和不同的模型搭配上面。

3.原先有試著找外部資料，例如美股、台股加權指標和美元匯率當做新的特徵，無奈 TEJ 軟體在比賽期間有缺少資料的情況(星期日等資料只看到上周四的匯率或者是一些台股加權指標)，最後放棄外部資料這部分，只採用原始資料來做預測。

環境

[系統環境]Win10 x64
[程式環境] Rstudio Version 1.1.383
[處理器]Intel(R) Core(TM) i7-3770

特徵

所有個股資訊(開盤、收盤、最高值、最低值、交易張數)、除了欲預測的 ETF 剩餘 17 支基金的資訊(開盤、收盤、最高值、最低值、交易張數)、欲預測的 ETF 基金的資訊(開盤、最高值、最低值、交易張數)、當天為星期幾。總共有 8541 個特徵。
(以上只使用調整後的資料)

訓練模型

使用了 OGA+HDAIC、Xgboost、SVM、GLM 和 Random Forest 等模型，這些模型分別對應到 R 套件 Ohit、xgboost、e1071、glmnet、randomForest。其中比較陌生的是第一個 OGA+HDAIC 模型，這個模型對高維度的選模具有相當大的益處，具有 sure screening 和增加預測準確度的目標，可以參考 2011 年 C.-K Ing and T. L. Lai 在 Statistica Sinica 的 paper。Xgboost 和 Random Forest 這種 tree 形式的模型並沒有給定重要特徵，而是讓每棵樹依照各別的損失函數去尋找最佳的特徵。

參考資料:

OGA+HDIC+Trim (只取前兩階段):
http://mx.nthu.edu.tw/~cking/pdf/IngLai2011.pdf

XGboost
http://xgboost.readthedocs.io/en/latest//model.html

SVM
https://c3h3notes.wordpress.com/2010/10/20/r%E4%B8%8A%E7%9A%84libsvm-package-e1071/

GLM
https://en.wikipedia.org/wiki/Generalized_linear_model
https://www.r-bloggers.com/generalised-linear-models-in-r/

訓練方式及原始碼

1.Stacking 第一層
每一支基金均使用 5-fold CV 來調整每個模型的調整參數:
(1) OGA+HDAIC
使用 OGA 的方法快篩出重要的特徵，再利用 HDAIC 來去除多餘的特徵。
(2)Xgboost
不預設重要特徵，使用 Gradient Boosting 的方式選出最好的特徵。
(3) SVM
不預設重要特徵，使用全部特徵。(這邊是使用 regression 版本的 SVM)

2. Stacking 第二層

使用第一層的 4 個模型的預測值($\hat{y}_1, \hat{y}_2, \hat{y}_3, \hat{y}_4$)與真實值 y 當做新的 X 與 Y，比較不一樣的地方是($\hat{y}_1, \hat{y}_2, \hat{y}_3, \hat{y}_4$)的計算方式採用類似比賽模式對歷史資料做出預測值，因此每次會做出 90*4 的矩陣，並選擇最近 22 期(一期 5 天)當新的資料會得到一個 1980*4 的矩陣，這個做法優點是方便更新每周的資料不需要每次都重跑，省下非常多的時間，code 得部分這裡的 X 相當於 E1，Y 相當於 bb。最後配適 Xgboost、SVM、GLM 模型，這邊特徵只有 4 個所以忽略選出重要特徵的問題直接配適模型。

3. Stacking 第三層

利用第二層 3 個模型預測值($\hat{\hat{y}}_1, \hat{\hat{y}}_2, \hat{\hat{y}}_3, \hat{\hat{y}}_4$)與真實值 y 當做新的 X 與 Y，配適 Xgboost 模型。

4.由於得出結果為當周的漲幅表，需要經過轉換才能得到最終的預測值，詳見”ETF 比賽漲幅換算”的 excel 表格。

結論

這次競賽在 response 和模型選擇上著墨了很久，一開始有考慮因應分數的計算分別採用預測漲幅和漲跌符號，鑒於相關知識的不足以及時間上的考量，最後還是選擇只預測漲幅。模型的調整參數部分花費了非常龐大的時間(特別是 XGboost)，經過幾周的調整後，參數固定下來問題就不大。第二個非常耗時的部分是在前一部分提到的 E1 矩陣，原本採用 Stacking 第一層預測好的值直接配模型，滿多次都趕在最後一天才弄好預測值，後來才想出上一部份的方法，不僅在精確度有稍微的提升，最重要的是每周更新速度會快上很多，最後二周都只花了一個小時以內就得到預測值。