AIFT期末專題報告-第四組

A1105505 林彧頎 A1105524 吳雨宣

A1105545 潘妤揚 A1105549 杜佩真

一、介紹

好的選股模型可以幫助投資人篩選出較佳的投資股票組合。現狀存在的問題是普遍的選股模型沒有很高的平均年複利，為此我們研究16個屬性、年份加上模型參數的排列組合挑戰找出高平均年複利的訓練模型。

基於本研究，可以得知屬性、年分、模型參數對於優化選股模型有很大的影響。未來或許可以以這些當基礎，再訓練多一點屬性、訓練多一點年分提升模型的平均年複利。

二、動機

此專題的研究目標在於優化選股模型。依題目的需求，我們分別使用KNN、Decision tree、Gaussian Naive Bayes model當作選股模型的基礎。以下是各個模型在初始狀態的設定

KNN: 屬性組合:全部屬性、train data:1997

Decision tree: 屬性組合:全部屬性、train data:1997

Gaussian Naive Bayes model: 全部屬性、train data:1997

以下是調整過後的設定

KNN: (此為某一種共有十種要分析)k=41、屬性組合: '股價淨值比', '股價營收比', 'M營業利益成長率', 'M稅後淨利成長率'、train data:1997, 1998, 2000, 2001, 2006

Decision tree: max\_depth=2、ccp\_alpha=0.025、屬性組合: '股價營收比', '利潤邊際NPM', 'M速動比率'、train data :1997, 1999, 2002, 2004, 2007

Gaussian Naive Bayes model: 屬性組合: '本益比', 'M淨值報酬率─稅後', 'M應收帳款週轉次','M營業利益成長率', 'M稅後淨利成長率'、train data : 1997, 1999, 2007

此外，我們有額外新增一個屬性EPS，使得KNN模型的表現可以更好。

三、專題研究(測試資料的結果於前三題各自的最後一大題)

1.KNN

**1.KNN介紹與使用方法:**

KNN是依據最近的k個鄰居所做分類。KNN可以用於分類與回歸問題上，並在分類未知的資料時，此方法計算出未知的資料與訓練資料之間的距離，並且找出最近的k個點，再以多數決的方式分類。而KNN一旦處理龐大的資料，分類速度就會變得緩慢，無法順利學習高維度的資料。因為KNN再分類未知的資料時，必須對大量的訓練資料做搜尋，找出鄰近的點。

* **KNN主要的方法:**

a.猜測法: 不斷地猜總會猜到正確答案。

b.啟發法: (有三種方法)1.使用類別與k互質的組合。2. K值需大於或等於類別數加一。3. 選擇低至足以避免雜訊的k值。

c. 優化演算法: 利用基因演算法(Genetic algorithm)或是網格搜尋法 (Grid search)暴力地找尋最佳解。

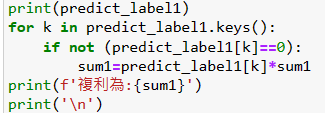
本研究目前使用啟發法中選擇低至避免雜訊的k值，找出錯誤率最低的k值進行訓練。

* **KNN使用方法:**

一開始先儲存所有的變數資料，預測的作法為：計算所有資料點到x的距離，隨著遠離x的距離開始分類資料，最後再預測k值範圍內最主要的標籤。優點在於非常簡單，適用於任何類別，需要的參數少。缺點在於高預測費、不適用多維度資料及類別變數。

但是可以從中發現KNN主要適用於分類，如何找出最佳的股票，需要想不同的方法。

我們這裡主要是將所有樣本中，距離由小到大排序，排序後找出相對的位置，並且因為knn分群的結果中，最小的距離可能是負的預測，因此我們這裡又篩選出為正的預測的股票，而另外其實我也有實作另一種方法可以透過K值抓的股票數中predict為1的label做return運算，也有不錯的效果，底下將有更詳細的說明。



圖表 1 利用k值選股數

**2.本研究的解決方法:**

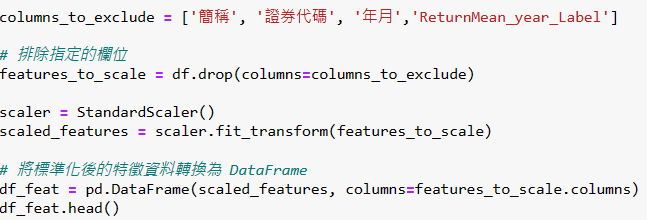
**本研究的資料夾請參考第一題**

**A.找最佳屬性:**

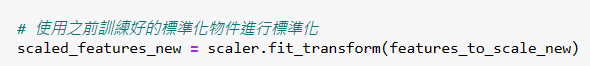
**(本程式請參考找最佳屬性的資料夾:1.全部屬性為測試2.部分屬性為找最佳屬性)**

首先我在寫程式的過程中，先找最佳的屬性值，而由於我們的時間有限，我大概在專題報告前2~3週，就開始向助教與老師詢問關於專題實作的要求，那時候的我以為要求是研究關於屬性的組合可以產生最好的return為主要目標，而我知道這件事之後，我馬上就開始動工，首先我先引用python中，有提供快速計算KNN的函數，計算KNN，主要計算過程為:

(1). 儲存所有的變數資料(包含訓練的資料與測試的資料):這裡需要做到資料處裡，因為最後的欄位return的label與前面說明股票日期、名稱等等…並不是我們拿來做測試的資料，因此需要剃除。



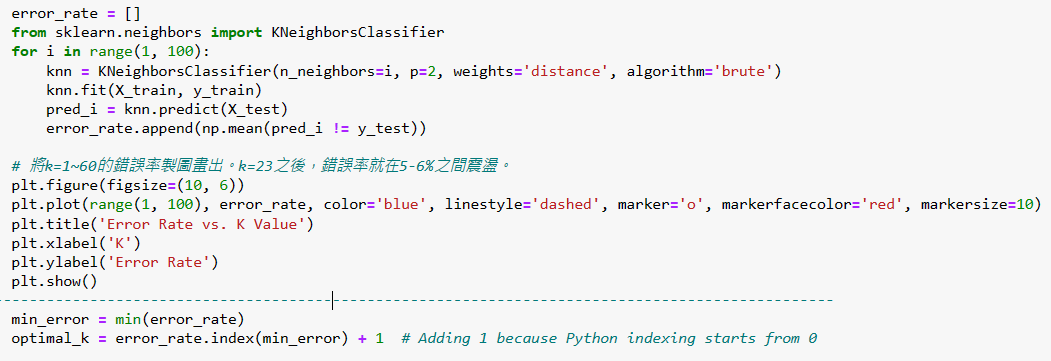
(2). 正規化資料:因為有資料不平衡的問題，因此需要做正規化資料。



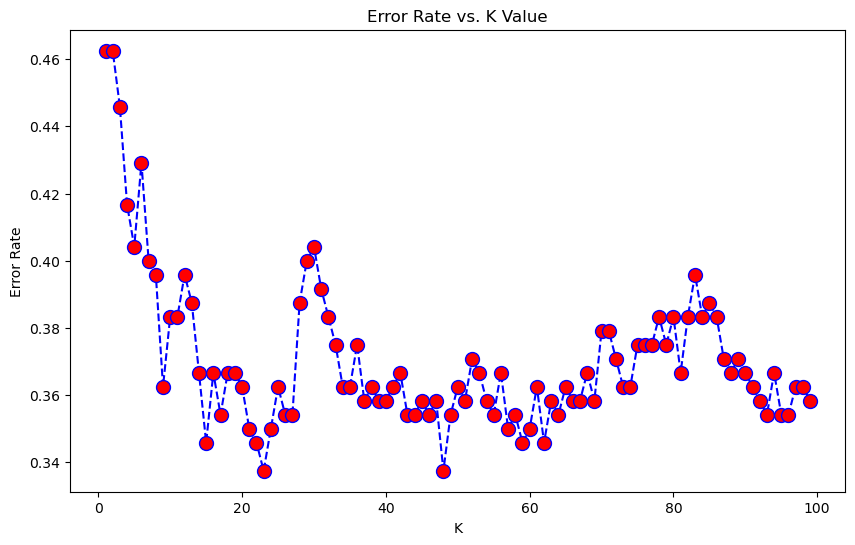
(3). 計算所有資料點到x的距離: 隨著遠離x的距離開始分類資料



(4). 預測k值:我們在這裡是使用選擇低至避免雜訊的k值，找出錯誤率最低的k值進行訓練。



(以下是範例圖)



以上訓練完後，我們就會找到最佳的k值，接著我寫了一個程式碼，可以動態產生所有屬性的組合，因此我大約跑了**1天~2天測試所有屬性的組合**，而這時候計算出來的公式是尚未進行年均化的複利值，以下是我們的算法:

(每年的前幾股相加求均值(會介於-1~1之間))+1，此為一年的報酬，再將每年的報酬相乘，會得到複利值。

而經過統計得出的十大組合-前十大複利值，我研究出最高可達37.52，而以下為我研究出的前十大組合:

(以下資料請參考資料夾test\_Attribute)

前十大組合:

**(index:屬性組合、最佳k值、複利值)**

第一大:

810:[['股價淨值比', '股價營收比', 'M營業利益成長率', 'M稅後淨利成長率'], 41, 37.52339415172343]

第二大:

5171:[['股價淨值比', 'M淨值報酬率─稅後', '資產報酬率ROA', 'M存貨週轉率 (次)', 'M應收帳款週轉次', 'M稅後淨利成長率'], 19, 17.00445257846897]

第三大:

5530:[['股價淨值比', '利潤邊際NPM', 'M存貨週轉率 (次)', 'M應收帳款週轉次', 'M營業利益成長率', 'M稅後淨利成長率'], 74, 15.098141450860702]

第四大:

5957:[['股價營收比', '營業利益率OPM', 'M存貨週轉率 (次)', 'M應收帳款週轉次', 'M營業利益成長率', 'M稅後淨利成長率'], 4, 13.963183234088522]

第五大:

6507:[['本益比', '股價淨值比', '股價營收比', 'M淨值報酬率─稅後', '資產報酬率ROA', 'M存貨週轉率 (次)', 'M營業利益成長率'], 47, 13.785077266970829]

第六大:

8257:[['股價淨值比', '股價營收比', 'M淨值報酬率─稅後', '資產報酬率ROA', 'M流動比率', 'M速動比率', 'M應收帳款週轉次'], 10, 13.597424715968453]

第七大:

13185:[['本益比', '股價淨值比', '股價營收比', '資產報酬率ROA', '營業利益率OPM', '利潤邊際NPM', 'M流動比率', 'M應收帳款週轉次', 'M營業利益成長率'], 35, 12.910328285691815]

第八大:

15975:[['本益比', '股價淨值比', '股價營收比', 'M淨值報酬率─稅後', '資產報酬率ROA', '利潤邊際NPM', '負債/淨值比', 'M流動比率', 'M速動比率', 'M營業利益成長率', 'M稅後淨利成長率'], 55, 12.466502520715848]

第九大:

9541:[['股價營收比', '營業利益率OPM', '負債/淨值比', 'M速動比率', 'M存貨週轉率 (次)', 'M營業利益成長率', 'M稅後淨利成長率'], 7, 12.35137173447864]

第十大:

16076:[['本益比', '股價淨值比', '股價營收比', '營業利益率OPM', '利潤邊際NPM', 'M流動比率', 'M速動比率', 'M存貨週轉率 (次)', 'M應收帳款週轉次', 'M營業利益成長率', 'M稅後淨利成長率'], 48, 8.650500458742039]

並且我將這些前十大組合的屬性計算出現的頻率，我發現基本上第一名的屬性最常出現在這些組合之中，但我最終取股價淨值比、股價營收比、M淨值報酬率─稅後、M營業利益成長率, M稅後淨利成長率，做屬性方面的第二階段測試。

屬性頻率:(找前幾名中的組合計算頻率)



(以下資料請參考資料夾測試training資料組合)

(1.依照屬性頻率測試: KNN\_year\_attribute)

(2.依照top10組合測試: KNN\_year\_attribute\_top10)

**B.測試training資料組合**

到了報告前一週，我才了解原來老師pdf上的TV的意思是要找出最好的training年份組合，因此我趕緊趕工將關於年份的組合產出，並修改程式碼，並延伸研究方向，我在程式中，不僅僅讓他可以研究哪一年的組合做為training資料最好，並且也搭配前面找出的十大屬性組合與前幾名頻率的屬性找出最好的年均化複利，但這些方法十分耗空間與時間，我跑這些資料，**不眠不休跑了快5~6天**，由於跑程式的過程中，很難精進程式碼，因此我在一開始徹底跑之前，就必須先將程式碼徹底檢查並研究過，我才可以好好地跑，另外為了**跑將近20萬以上的數據**，我同時並行處理多個程式，因此我建了很多程式並行執行，另外我也為了不要讓數據錯誤，我可以說是，前期處裡程式的時候，我就將程式從頭寫到最後要提供給助教測試testing的資料的程式碼，並且每寫完一個就去檢查另一個，使時間可以再被壓縮，希望可以求出最好的值。

以下是第二階段主要的方法步驟:

(0). 將所有年份的組合產生:須注意的是當training資料有A時，則testing的資料為全部年份減A所剩下的年份。



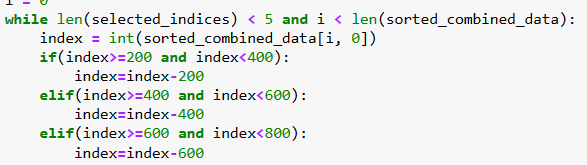
(1). 儲存所有的變數資料(包含訓練的資料與測試的資料):這裡需要做到資料處裡，因為最後的欄位return的label與前面說明股票日期、名稱等等…並不是我們拿來做測試的資料，因此需要剃除，另外將所有要測試的屬性進行指定，處理變數資料。(程式碼大致如A所示)

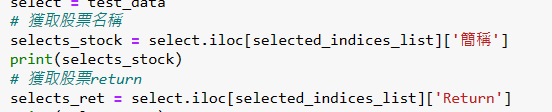
(2). 正規化資料:因為有資料不平衡的問題，因此需要做正規化資料。(程式碼大致如A所示)

(3). 計算所有資料點到x的距離: 隨著遠離x的距離開始分類資料。(程式碼大致如A所示)

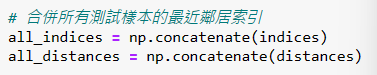
(4). 預測k值:我們在這裡是使用選擇低至避免雜訊的k值，找出錯誤率最低的k值進行訓練。(程式碼大致如A所示)

(5). 資料量不同處理:本來原始code，training資料每年可能僅需要比200資料，而現在training資料變多，需要有相對應的分配，我這裡是用相對位置，並處裡資料的方法以抓取正確的資料return。





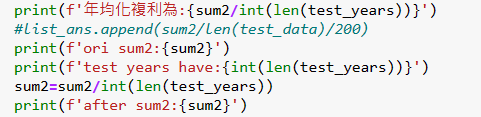
(6). 將前面訓練完的模型，給testing資料測試，進行分類並找出距離。



(7). 透過距離排出:距離由小到大排序，排序後找出相對的位置，並且因為knn分群的結果中，最小的距離可能是負的預測，因此我們這裡又篩選出為正的預測的股票。



(8) 計算年均化return: (每年的前幾股相加求均值(會介於-1~1之間))+1，此為一年的報酬，再將每年的報酬相乘，會得到複利值，而此複利值/測試年份數=年均複利值。



最後我們完成KNN研究方面的測試。

(以下是我們測試出來比較好的前幾個組合(有放在result測試結果的資料夾與程式碼中)



**C.實作測試用模型**(請參考RESULT\_TEST資料夾)

這個步驟其實就是將前面的模型訓練做修改，讓他可以依照我們指定的資料、訓練與測試年分、挑選的股數與預測式的屬性組合進行選股並提供時計算出的年均複利值。以下是方法: (以下CODE請參考上面所述)

(1). 儲存所有的變數資料(包含訓練的資料與測試的資料):這裡需要做到資料處裡，因為最後的欄位return的label與前面說明股票日期、名稱等等…並不是我們拿來做測試的資料，因此需要剃除，另外將所有要測試的屬性進行指定，處理變數資料。

(2). 正規化資料:因為有資料不平衡的問題，因此需要做正規化資料。

(3). 計算所有資料點到x的距離: 隨著遠離x的距離開始分類資料

(4). 預測k值:我們在這裡是使用選擇低至避免雜訊的k值，找出錯誤率最低的k值進行訓練。(以下是範例圖)

(5). 資料量不同處裡:本來原始code，training資料每年可能僅需要比200資料，而現在training資料變多，需要有相對應的分配，我這裡是用相對位置，並處裡資料的方法以抓取正確的資料return。

(6). 將前面訓練完的模型，給testing資料測試，進行分類並找出距離。

(7). 透過距離排出:距離由小到大排序，排序後找出相對的位置，並且因為knn分群的結果中，最小的距離可能是負的預測，因此我們這裡又篩選出為正的預測的股票。

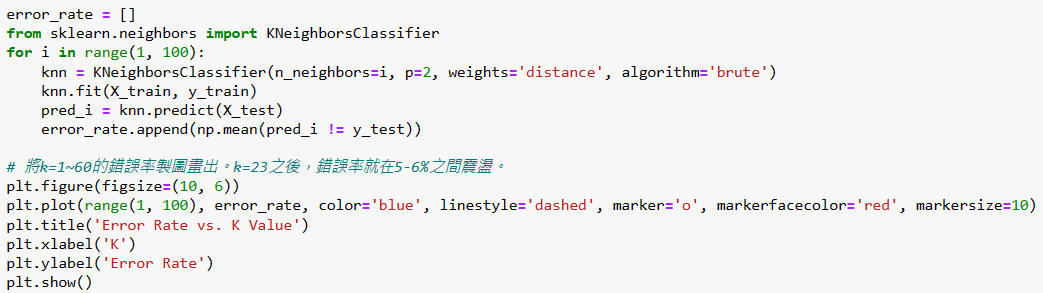
(8) 計算年均化return: (每年的前幾股相加求均值(會介於-1~1之間))+1，此為一年的報酬，再將每年的報酬相乘，會得到複利值，而此複利值/測試年份數=年均複利值。

總體來說，我在研究中，可以說是時做了3星期的研究，我最終也可以發現年均複利可以高達5.02以上，雖然或許不是最高，但某種程度上，如果每年都可以有這種報酬產生，則此模型應該可以說是具有一定的成功度，但我也有去網路上做一些其他研究，雖然不會作為我主要測試的資料，但也是可以幫助我評估其他方法訓練的模型。

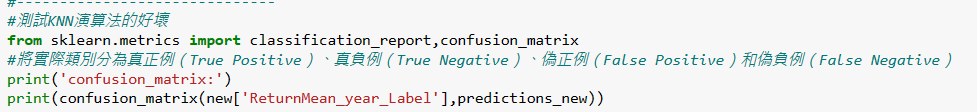
**D.其他的研究**

我有看到網路上有論文研究使用SVM+KNN的方法，但我發現他們主要是用來做比較數據，很少是拿來選股，因此對我來說選股依舊是一大挑戰，為了證明我的研究的可行信(準確度等等…)，我寫了一些程式碼以作為判斷:

(1). K值對應的錯誤率判斷圖:找出最佳的K值。



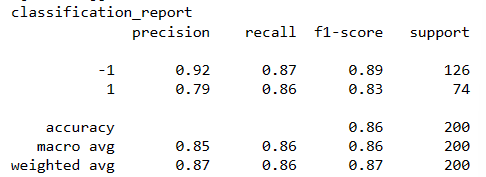
(2). Confusion matrix:依照老師ppt中教的分成，真正例（True Positive）、真負例（True Negative）、偽正例（False Positive）和偽負例（False Negative）。





(3). 模型的精確度、召回率、F1分數和支持數等指標，用來評估模型對於每個類別的預測性能



****

3.結論

首先，我研究出Top10組中最好的值為3.9，發生於5股時，他的屬性是: ['股價淨值比', 'M淨值報酬率─稅後', '資產報酬率ROA', 'M存貨週轉率 (次)', 'M應收帳款週轉次', 'M稅後淨利成長率']，K值是61，而其中一個有趣的現象是，當股數上升往往會有很大的落差，但第四名與第五名卻可以維持在一個不錯的水平之上，我覺得這應該與測試資料量有關，因為這些的測試資料都相對少，訓練資料多，雖然無法達到很好的return值，卻可以維持於一個水平之上。

再來是前五大屬性頻率的組合產生的結果，產生更高的年均化複利值5.02，並且個股最高階發生於此，以下是有關於此數據的屬性: train\_year:[1997, 1999, 2004, 2007],test\_year:[1998, 2000, 2001, 2002, 2003, 2005, 2006, 2008]:屬性為['股價淨值比', '股價營收比', 'M淨值報酬率─稅後'], K值為96。這些屬性可以說剛剛好是前面測試中最常出現在前幾名的組合的屬性，我想這也是為甚麼這個組合可以產生最好的return。

另外在屬性中，都可以維持在一個不錯的水平之上，我覺得這應該與測試資料量和屬性皆為比較好的可能性有關，因為這些的測試資料都相對少，訓練資料多，雖然無法達到很好的return值，卻可以維持於一個水平之上。

最後如果要找最好的10股、20股、30股也可以透過前5大屬性組合的第一名做挑選!

透過這項研究，我可以發現，KNN其實主要是用於分類，如果要做到最好的選股，或許需要有更深入的研究，但我對我這幾個星期，跑了這麼多筆資料並沒有感到失望，因為我最終找到了很不錯的年均化複利值，也訓練好模型，雖然測試時，有可能會有比較不好的結果，但這有時候會是跟突出較好的資料有關，也有可能是測試資料比較差，因此return不一定好。

但總而言之，可以看出KNN選股策略與分群方法，並對於股市選股具有一定可靠性與準確度!

(下一頁有結果圖表)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 第一名 | | | | 第二名 | | | | 第三名 | | | |
| 5  股 | 10股 | 20股 | 30股 | 5  股 | 10股 | 20股 | 30股 | 5  股 | 10股 | 20股 | 30股 |
| Top10  組合 | 3.9 | 1.6 | 1.53 | 0.94 | 3.6 | 1.1 | 0.64 | 0.53 | 3.5 | 1.9 | 1.1 | 0.9 |
| 前5大  屬性  組合 | 5.0 | 2.4 | 1.9 | 1.59 | 4.5 | 2.3 | 1.2 | 1.1 | 4.3 | 1.4 | 1.5 | 1.3 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 第四名 | | | | 第五名 | | | |
| 5  股 | 10股 | 20  股 | 30股 | 5  股 | 10股 | 20股 | 30股 |
| Top10  組合 | 3.4 | 2.1 | 1.5 | 1.3 | 3.3 | 1.9 | 1.1 | 1.3 |
| 前5大  屬性  組合 | 4.2 |  | | | 3.9 |  | | |

**4. 測試資料結論**

|  |  |
| --- | --- |
| **Training year** | **[1997, 1999, 2004, 2007]** |
| **Testing year** | **[2010,2011,2012,2013,2014,2015]** |
| **Training Attributes** | **['股價淨值比', '股價營收比', 'M營業利益成長率', 'M稅後淨利成長率']** |
| **年均化報酬率** | **0.308(30.8%)** |

此結果相對於training來的低是因為，training中有幾年會有好的return，而2010~2015年的股市相對較差，另一個原因是因為training資料是經過大數據分析後最好的幾個結果，而測試資料相對來說並沒有加入大數據分析，僅能依照training得出的屬性與年份來獲得預測，並測試結果，但整體來說，雖然年均化複利並沒有training高，但最終的年均化複利也表示我們的選股與策略是會收益為正，而且可以每年有穩定的報酬!

2. Decision Tree

Decision tree

Decision tree會根據訓練資料生成一棵樹，利用這棵樹可以用來預測新的資料。決策樹是以貪婪法則來決定每一層要問的問題，目標是分類過後的每一群可以明顯地知道屬於哪個類別。決策樹演算法可以使用不同的指標來評估分支的好壞，常見的為information gain。Entropy是計算information gain的一種方法，**Entropy=-**，**information gain=-p\*lgp-q\*lgq**(p:是的機率，q:否的機率) 決策樹演算法是採用 **CART** (Classification and Regression Tree) 演算法，每一個節點只分枝出兩個子節點，並且不斷地往下拓展，直到設定的最大深度為止。

程式設計

1. 找出決策樹的參數

採用entropy方法

max\_depth:

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

ccp\_alpa:

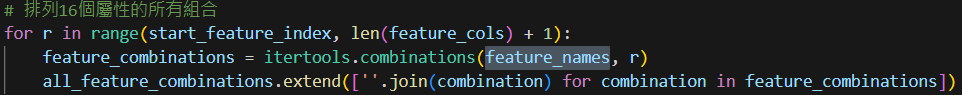
一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

1. 測試16個屬性組合

設定16個屬性為ABCDEFGHIJKLMNOP，列出屬性的各種組合，用2007年訓練模型，並計算前五個最有可能是1的股票，算出平均年複利，找出最高的平均年複利，記錄他的屬性組合以及準確率。





一張含有 文字, 字型, 螢幕擷取畫面, 數字 的圖片

自動產生的描述

1. 測試1997年至2008年份組合

列出1997年至2008年的各種組合，利用上述求出的屬性組合、參數訓練模型，並計算前五名最有可能是1的股票，算出平均年複利，找出最高的平均年複利，記錄他的年份組合。

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述





一張含有 文字, 字型, 螢幕擷取畫面, 數字 的圖片

自動產生的描述

研究說明

一張含有 文字, 圖表, 繪圖, 行 的圖片

自動產生的描述

此圖說明決策樹的深度最好設為2

一張含有 文字, 圖表, 螢幕擷取畫面, 繪圖 的圖片

自動產生的描述

此圖說明決策樹的ccp\_alpha最好設為0.0225至0.030之間

所以我的ccp\_alpha取0.025







上圖說明使用2007年的'股價營收比', '利潤邊際NPM', 'M速動比率'、深度設為2、ccp\_alpha設為0.025可以得到最高的前五名最有可能是1的股票的平均年複利約為1.3893





上圖說明使用1997、1999、2002、2004、2007年的'股價營收比', '利潤邊際NPM', 'M速動比率'、深度設為2、ccp\_alpha設為0.025加以訓練可以得到平均年複利約為7.0287

結論

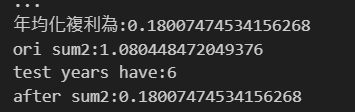




最終將以'股價營收比', '利潤邊際NPM', 'M速動比率'作為最佳屬性組合進行預測。

測試結果

測試2010-2015年的結果如下:

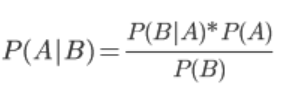


3. GaussianNB

**這個演算法在幹嘛?**

貝式分類器 (Bayesian Classifier) 是一種基於機率模型的機器學習模型，並且根據貝氏定理（Bayes' theorem）為基礎，透過機率統計來判斷未知的資料類別，常用的貝氏分類架構: 高斯貝式分類器 (GaussianNB)、多項式貝氏分類器 (MultinomialNB)、伯努力貝氏分類器 (BernoulliNB)。

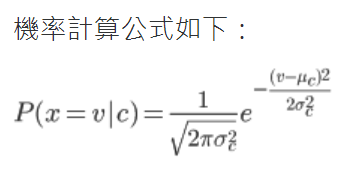
貝氏定理:



高斯貝氏分類器 (GaussianNB)：

高斯貝氏分類器是貝氏分類器的一個變體，主要用於處理連續型特徵。

它基於高斯分布（正態分布）的假設，用於估計特徵的概率分布，並且常用於處理數值特徵。



(首先對資料進行類別分類，假設訓練集中有一個連續屬性，然後計算每個類別中 的均值和變異數。令為在c類上的均值，令為在c類上的變異數。)

**怎麼解的?**

AI\_finance\_final 資料夾\_Q3\_高斯貝式分類器 (GaussianNB).ipynb:

1. 以1997年的資料作為訓練集，建立並訓練Gaussian Naive Bayes model



1. 以1998-2009年的資料做為測試集，對數據進行預測:

在內層的年度迴圈中，讀取每年的數據 (top200\_{year}.xlsx)，並標準化特徵。然後，使用訓練好的Gaussian Naive Bayes model對數據進行預測



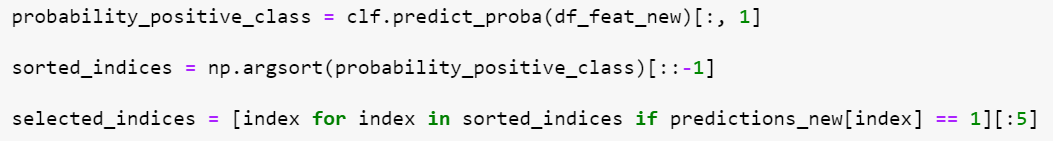
1. 評估模型性能:

使用混淆矩陣、分類報告等方法評估模型對新數據的預測性能。

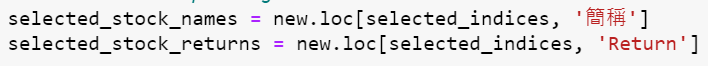
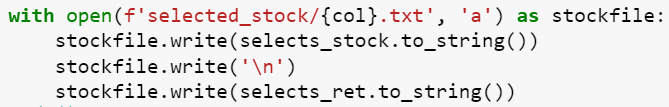
1. 選擇預測為1的前5隻股票:

使用 predict\_proba 方法獲取對正類別（1）的概率估計，然後根據概率降序排列實例的索引。選擇預測標籤為1的前5個實例。

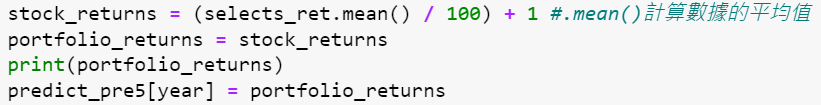


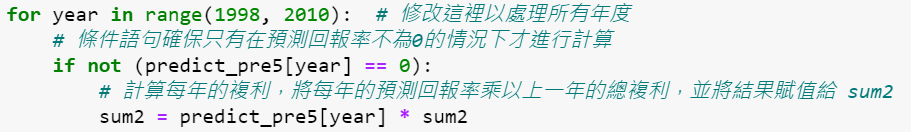
1. 將選中的股票寫入文件:

根據選中的索引，從原始數據中獲取相應的股票名稱和Return，並寫入文件。

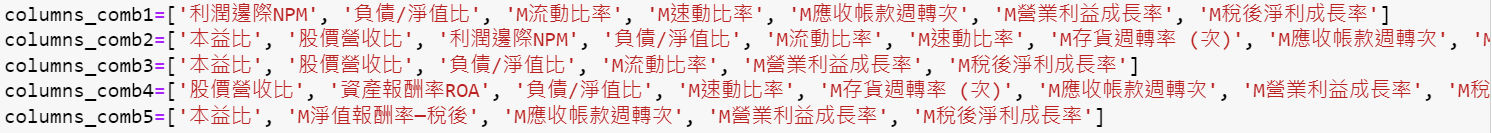
1. 計算複利(非年均化)



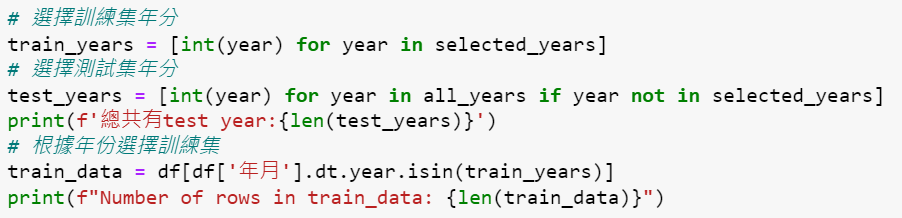


Q3\_測試前5大屬性 資料夾\_ Q3\_test5attribute.ipynb:

1. 獲取年份組合並存入'year\_combinations.txt'，從’AI\_finance\_final’所生成的’test1.txt’觀察複利(非年均化)前5高的屬性組合為何



1. 獲取年份組合有些作為訓練集，其餘做為測試集



1. 建立並訓練Gaussian Naive Bayes model



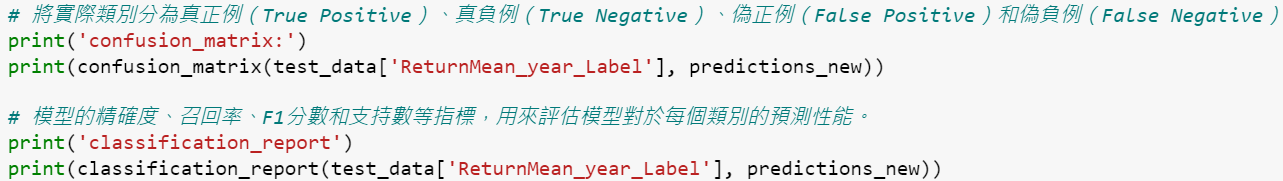
1. 以年份組合和前5高的屬性組合對數據進行預測:

使用訓練好的Gaussian Naive Bayes model對數據進行預測



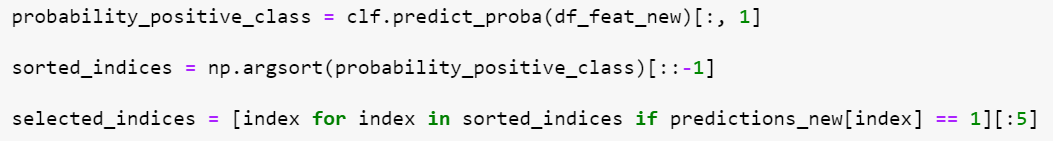
1. 評估模型性能:

使用混淆矩陣、分類報告等方法評估模型對新數據的預測性能。



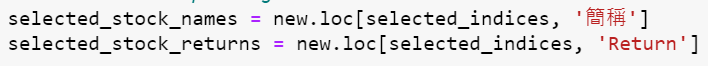
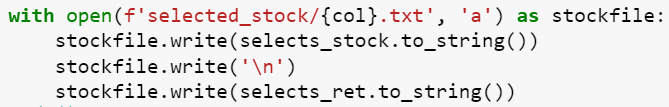
1. 選擇預測為1的前5隻股票:

使用 predict\_proba 方法獲取對正類別（1）的概率估計，然後根據概率降序排列實例的索引。選擇預測標籤為1的前5個實例。

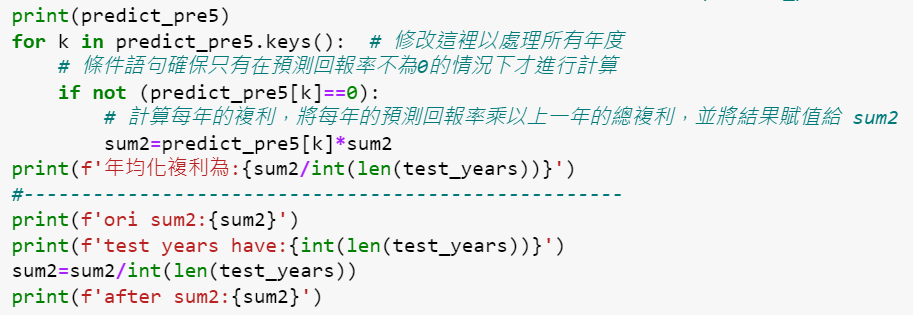


1. 將選中的股票寫入文件:

根據選中的索引，從原始數據中獲取相應的股票名稱和Return，並寫入文件。

1. 計算複利(非年均化與年均化)



Result\_test\_Q3 資料夾\_ GaussianNB\_EVAL\_RESULT.ipynb:

1. 獲取年份作為訓練集與測試集





1. 建立並訓練Gaussian Naive Bayes model



1. 以年份組合和選擇的屬性組合(基於’Q3\_測試前5大屬性’的結果，找出最佳屬性)對數據進行預測:

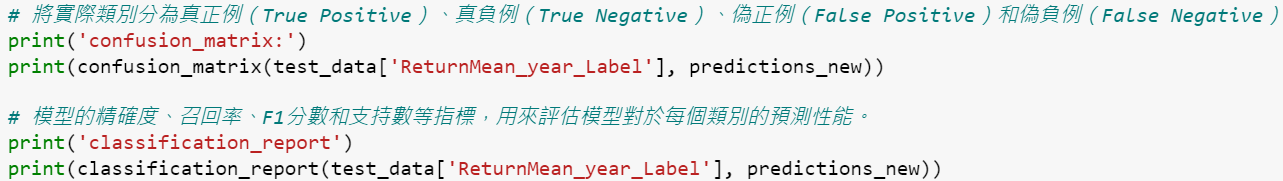
使用訓練好的Gaussian Naive Bayes model對數據進行預測





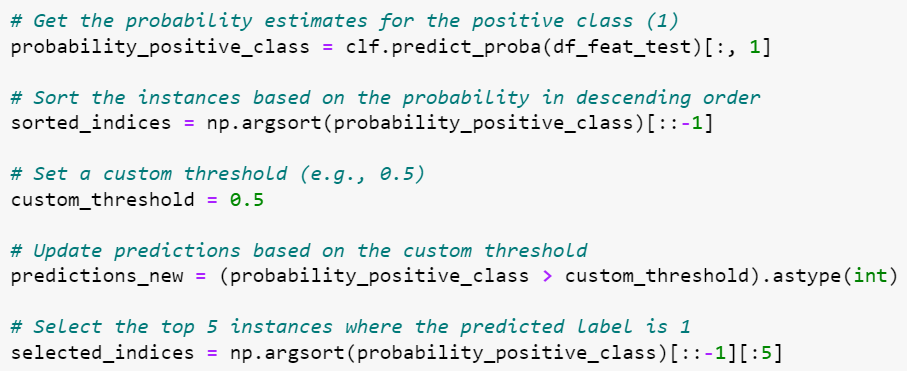
1. 評估模型性能:

使用混淆矩陣、分類報告等方法評估模型對新數據的預測性能。



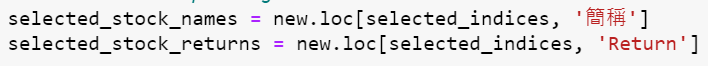
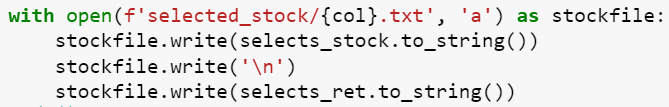
1. 選擇預測為1的前5隻股票:

使用 predict\_proba 方法獲取對正類別（1）的概率估計，然後根據概率降序排列實例的索引。選擇預測標籤為1的前5個實例。

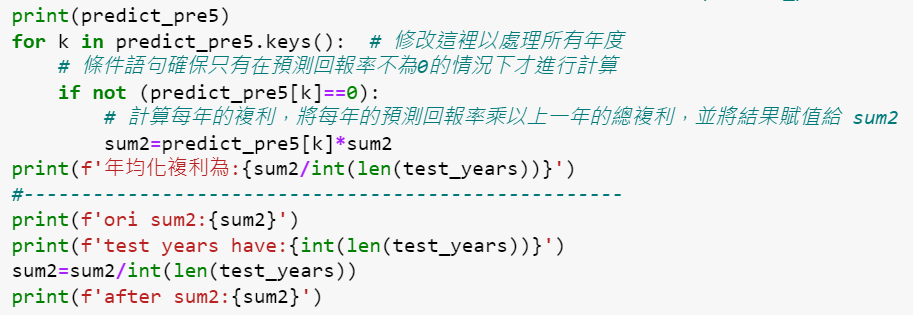


1. 將選中的股票寫入文件:

根據選中的索引，從原始數據中獲取相應的股票名稱和Return，並寫入文件。

1. 計算複利(非年均化與年均化)



**訓練結果:**

AI\_finance\_final資料夾\_Q3\_高斯貝式分類器 (GaussianNB).ipynb:

透過Gaussian Naive Bayes model訓練與預測的數據，當在預測時，所選取的股票數為5:

9903:[['利潤邊際NPM', '負債/淨值比', 'M流動比率', 'M速動比率', 'M應收帳款週轉次', 'M營業利益成長率', 'M稅後淨利成長率'], 6.816019081885784]

15572:[['本益比', '股價營收比', '利潤邊際NPM', '負債/淨值比', 'M流動比率', 'M速動比率', 'M存貨週轉率 (次)', 'M應收帳款週轉次', 'M營業利益成長率', 'M稅後淨利成長率'], 6.796577581828283]

4272:[['本益比', '股價營收比', '負債/淨值比', 'M流動比率', 'M營業利益成長率', 'M稅後淨利成長率'], 5.967644298226404]

12708:[['股價營收比', '資產報酬率ROA', '負債/淨值比', 'M速動比率', 'M存貨週轉率 (次)', 'M應收帳款週轉次', 'M營業利益成長率', 'M稅後淨利成長率'], 5.7185946704339585]

1975:[['本益比', 'M淨值報酬率─稅後', 'M應收帳款週轉次', 'M營業利益成長率', 'M稅後淨利成長率'], 5.476316556802925]

根據所選取的股票計算出來的投資複利最高的是前五個屬性組合為如上。

Q3\_測試前5大屬性 資料夾\_ Q3\_test5attribute.ipynb:

max:train\_year:[1997, 1999, 2007],test\_year:[1998, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2008]:478:[['本益比', 'M淨值報酬率─稅後', 'M應收帳款週轉次', 'M營業利益成長率', 'M稅後淨利成長率'], 1.527768308495194]   
根據所選取的股票計算出來的投資年均化複利最高的屬性組合如上。

**測試結果:**

Result\_test\_Q3 資料夾\_GaussianNB\_EVAL\_RESULT.ipynb:

以['本益比', 'M淨值報酬率─稅後', 'M應收帳款週轉次', 'M營業利益成長率', 'M稅後淨利成長率']作為測試的屬性組合，進行測試。

測試結果以1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009為訓練年，2010, 2011, 2012, 2013, 2014, 2015為測試年，最大年均化複利為0.3439。   
(結果如下)

max:train\_year:[1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009],test\_year:[2010, 2011, 2012, 2013, 2014, 2015]:[['本益比', 'M淨值報酬率─稅後', 'M應收帳款週轉次', 'M營業利益成長率', 'M稅後淨利成長率'], 0.34393496969186405]

4. Improve

1. 屬性選擇-EPS

在期中專題中我們有找了10我們覺得會影響股市的屬性，而期末專題中教授給了16個屬性。

在這16個屬性中，只有EPS是沒有在這些屬性中的，而這是我們在期中專題時認為會影響股市的因素之一，所以在第四題中我們選擇添加此屬性來試試看是否可以優化模型。

1. 模型選擇-knn

因為在前面三題中，我們發現knn所得到的報酬是三個模型中較佳的，所以我們在第四題選擇將添加的屬性應用在knn上，比較是否能得到更好的獲利值。

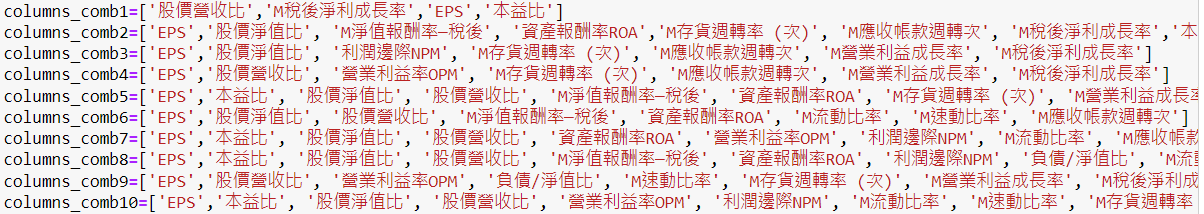
1. 方法(EPS\_KNN\_year\_attribute.py)

將訓練資料添加EPS的數據



使用第一題的方法，並且我們將第一題中挑選到的前十大屬性組合都添加EPS屬性，並去配對每一種年份組合。

另外，在訓練1997年時也有得出含有EPS數據時，較高的屬性組合，所以也有將這些屬性以及第一題添加EPS後的十大屬性互相配對，看是否可以提高獲利。



更改訓練年份範圍



更改屬性組合範圍



1. 結果

最高的年均化複利的訓練模型為以下：

訓練年份為：1997 1999 2002 2003 2007

|  |  |
| --- | --- |
|  | 年均化複利 |
| 屬性：  股價淨值比 淨值報酬率-稅後  資產報酬率ROA 存貨週轉率(次)  應收帳款週轉次 稅後淨利成長率 EPS 本益比 | 4.67639 |

1. 發現

我們發現雖然加入EPS後，獲利沒有比第一題得到的數據高，但或許是因為我們在添加EPS的過程中，因為有些股票找不到EPS的數據，所以我們會先直接刪除，這可能會影響也許return高的會因為沒有EPS數據而被刪掉，所以就不會被選到。

但在訓練數據以及測試數據中也會發現，在有被刪除數據的資料中，含有EPS的是最高複利的屬性組合，所以可以發現EPS的確可以提高模型的訓練。

四、Correlation

1997-1999相關係數

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1997 | 1998 | 1999 |
| 市值(百萬元) | 0.107 | 0.319 | 0.025 |
| 收盤價(元)\_年 | -0.196 | -0.083 | -0.208 |
| 本益比 | 0.069 | -0.090 | -0.053 |
| 股價淨值比 | -0.197 | -0.114 | -0.170 |
| 股價營收比 | -0.234 | -0.083 | -0.091 |
| M淨值報酬率─稅後 | 0.309 | 0.209 | 0.251 |
| 資產報酬率ROA | 0.315 | 0.081 | 0.261 |
| 營業利益率OPM | 0.151 | 0.037 | 0.220 |
| 利潤邊際NPM | 0.156 | 0.116 | 0.218 |
| 負債/淨值比 | -0.094 | -0.079 | 0.138 |
| M流動比率 | 0.129 | 0.085 | 0.013 |
| M速動比率 | 0.265 | 0.127 | 0.028 |
| M存貨週轉率 (次) | -0.032 | 0.111 | 0.158 |
| M應收帳款週轉次 | 0.075 | -0.047 | 0.171 |
| M營業利益成長率 | -0.071 | -0.048 | 0.244 |
| M稅後淨利成長率 | -0.117 | -0.052 | 0.132 |

**Correlation03-05:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 2003 | 2004 | 2005 |
| 市值(百萬元) | 0.071198 | 0.013760 | -0.027569 |
| 收盤價(元)\_年 | -0.284899 | 0.165535 | -0.153954 |
| 本益比 | 0.013487 | -0.091860 | 0.076176 |
| 股價淨值比 | -0.368252 | -0.006251 | -0.185286 |
| 股價營收比 | -0.224904 | -0.005799 | -0.098511 |
| M淨值報酬率─稅後 | 0.024542 | 0.165855 | 0.049289 |
| 資產報酬率ROA | 0.098867 | 0.166706 | 0.066529 |
| 營業利益率OPM | 0.105306 | 0.113372 | 0.011941 |
| 利潤邊際NPM | 0.114405 | 0.108748 | 0.066100 |
| 負債/淨值比 | 0.034196 | -0.036149 | -0.002953 |
| M流動比率 | 0.087704 | -0.020155 | -0.062273 |
| M速動比率 | 0.041879 | 0.062588 | -0.088753 |
| M存貨週轉率 (次) | 0.019790 | 0.017419 | -0.060152 |
| M應收帳款週轉次 | 0.186417 | -0.058788 | 0.103939 |
| M營業利益成長率 | -0.033811 | -0.059607 | 0.282039 |
| M稅後淨利成長率 | -0.016756 | -0.078200 | -0.006399 |



(空白部分為相關係數絕對值<0.1)

五、總結

透過本次的專題，我們分工合作，每個人負責一題，我們互相合作，有時候有些程式碼事可以參考的，組員就可以透過互相學習，讓事半功倍，並且可以讓各自在各自擅長的領域實作，或達到1+1大於2的效果，透過本次專題，我們不僅學會選股模型的設計與相關演算法，並學習如何演練於程式之中，更重要的是，我們學會如何結合金融與人工智慧，使可以模擬在投資的市場中，如何透過長期投資來獲得最大的複利。

總而言之，本次的專題雖然很趕，有些數據可以在做的更好，但以短期來說已經是很好的效果了，更重要的是過程中的想法，是花很久很久時間不斷地驗證，並且跑大數據才知道的成果，像是KNN就跑了將近20萬以上的數據!

六、心得

A1105505 林彧頎

在本次的專題中，基本上是我把KNN寫好後，丟到雲端，而組員有從中學習，這樣可以幫助大家更快進入狀況，也可以更有效率，我覺得這次的期末專題，某種意義上彈性很大，因為想做多少研究可以取決於自己，而且在沒有正確答案的情況下，時間也十分趕，因此我們的做法就是很早之前就開始製作，但重點是，整體撰寫程式需要在前期十分小心，因為數據很龐大，因此不能有任何閃失，我在這次專題中，不僅僅學到我負責的KNN，也學會更好的處裡資料與分析資料，而也由於第四題同學在評估過我們的成果後，選擇KNN，使我有部分的correlation的研究可以參考到同學的研究，但其實也只有部分有相關，真正可以導致有好的效果的屬性，其實correlation不一定高，更重要的是，我也學習到在做人工智慧的過程中，其實資料很常有隨機性，有時候幾個突出的最佳值，可能是不小心產生的，因此如果testing的結果不好，其實也很正常，我覺得這次的專題如果可以有更多時間，或許可以做的更好，但整體來說，我獨自撰寫程式的過程中，學習到很多，並也學會互相幫忙，因為這個專題，其實班上很多同學都來問我，我在過程中，其實也是在重新學習，總而言之，我學到很多，謝謝這學期的指教。

A1105524 吳雨宣

這次的期末專題跟期中一樣，都需要花很多時間跑數據，一開始我想很久，不知道要選哪個演算法來訓練模型，所以就從老師PPT上面列的Method來選擇，後來選擇了貝氏分類器裡面的高斯貝氏分類器，雖然貝氏分類器的核心定理貝氏定理，高中就學過了，但從來沒想過原來可以應用在訓練選股模型上面，總之，從這次的專題裡面，我獲益良多。在這堂課裡面我學會了很多關於金融的相關知識，以及如何讓金融與人工智慧進行結合，更知道了自主學習的重要性。

A1105545 潘妤揚

我在這次的專題當中花的心力比我想像的多，同時也學到了很多東西。一開始原本以為只要寫好程式碼就好，但訓練的過程當中花的時間比我想像的多很多，大概佔了百分之七十都在等程式跑完。其中我最熟悉的部份是我負責的Decision tree，調整參數、選擇屬性、選擇年分，不停地測試、訓練，優化模型，感覺就是在看誰的硬體設備比較好，在有限的時間內訓練出較佳的選股模型，希望之後可以更了解金融方面的資訊。

A1105549 杜佩真

在這次的專題中，發現影響股市的因素真的非常的多，而且最高的報酬也會因為不同的屬性組合配上不同的股票數選擇而有所不同，在分析上也是非常的複雜，但在這個專題也讓我更加了解在分析股票上可以往什麼方向分析。

七、工作分配

|  |  |
| --- | --- |
| A1105505 林彧頎 | 第一題KNN + merge |
| A1105524 吳雨宣 | 第三題 Gaussian Naive Bayes model |
| A1105545 潘妤揚 | 第二題 Dicision Tree |
| A1105549 杜佩真 | 第四題 分析屬性 |