

一一三、Python AI 金融交易策略应用(3): 初级班内容范例

初级班的课程内容针对金融交易的基本观念有较为深入的理论说明，我发现很多从事量化交易的朋友，都是具有 IT 背景的朋友。对于程序开发比较具有优势，但是一些金融理论的知识，就明显的不如金融背景的朋友。虽然程序量化的技巧很重要，但金融的领域知识是不能轻忽的。

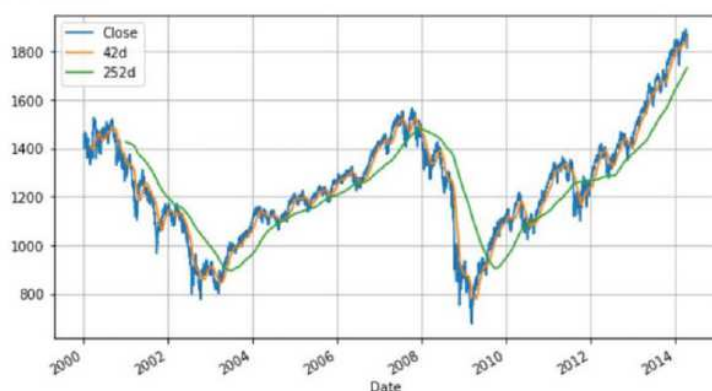
透过这些基本内容的学习，比较能深入交易的核心知识。我从初级班的内容中摘取三个主题，做了三页的说明，简单示范一下课程的内容。这些范例背后的程序都是课程说明的重点。搞量化的工作，最忌讳的事情是满口的理论模型，但实务上却是无法把代码实作出来。因此本课程的进行方式，是先介绍基本的理论知识，然后说明如何使用 Python 与相关的库，再把这些程序都实作出来，验证理论的说明。

下图是以长、短期均线进行趋势策略的应用，使用 252 日 MA 当长期趋势，42 日 MA 当短期趋势。当短期均线突破长期均线一定门坎后，进行买卖的动作。本程序把回溯测试的代码也一并实作出来，因此可以把讯号分布图与权益曲线一并呈现。

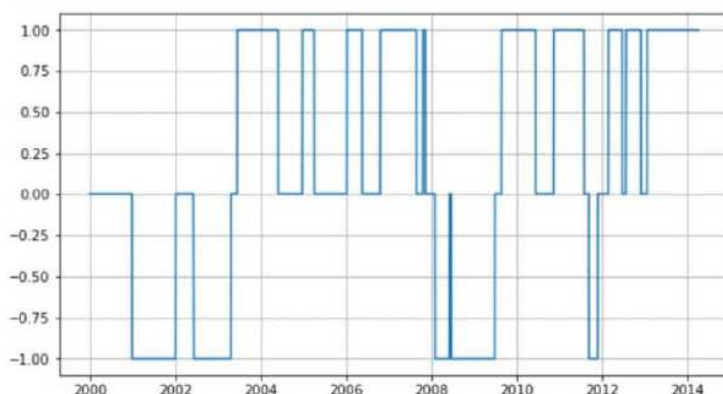
✧ 本課程介紹古典交易策略：使用長、短期平均線，進行趨勢策略的投資

◆ 畫均線，使用 pandas 功能

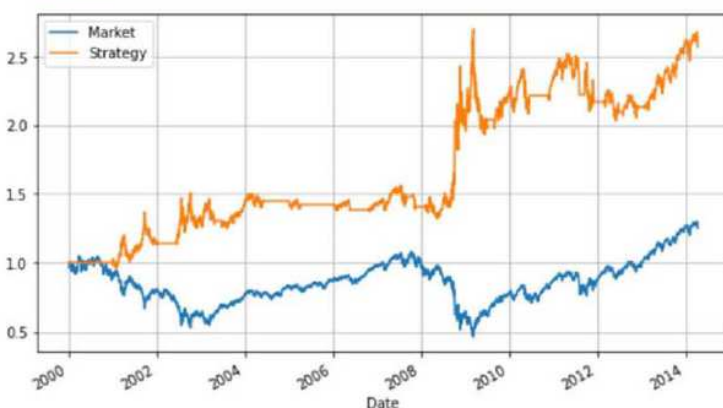
```
>>> sp500[['Close', '42d', '252d']].plot(grid=True, figsize=(18, 10))
>>> plt.show()
```



◆ 訊號分佈



◆ Equity Curve



下图是以蒙特卡罗模拟法搭配几何布朗运动过程，来仿真资产价格的路径。当我们拟好策略后，都需要进行交易测试。使用历史数据回测，往往受限于数据的有限性，无法进行大量测试。使用仿真法产生测试价格，便可突破这个限制。另外，模拟法也可用来定价期权，是一个很实用的方法。

◆ 本課程介紹交易策略驗證方法：使用蒙特卡罗模拟法進行策略驗證

◆ GBM 程序

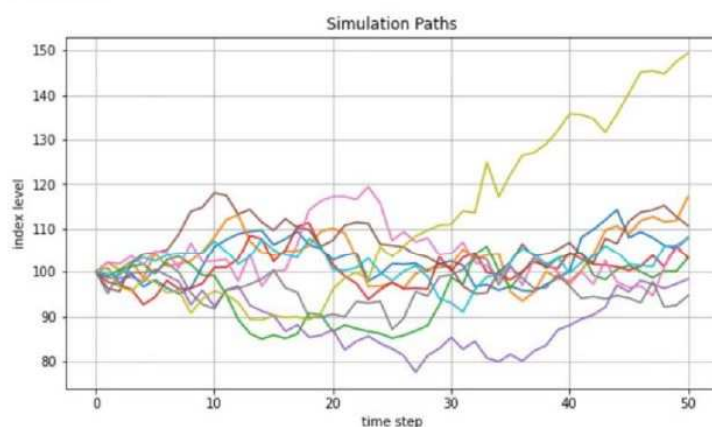
```
>>> t0 = time()
# Random seed
>>> random.seed(20000)

# Simulating I paths with M time steps
>>> S = S0 * exp(cumsum((r - y - 0.5 * sigma ** 2) * dt
>>> + sigma * math.sqrt(dt) * random.standard_normal((M, I)), axis=0))
```

◆ 選擇權價格

```
>>> C0 = math.exp(-r * T) * sum(maximum(S[-1] - K, 0)) / I
>>> print("European Option Value %.3f" % C0)
European Option Value 8.037
>>> t1 = time()
>>> print("Total Time: ", t1-t0)
```

◆ 股價模擬圖



◆ 到期股價分佈圖

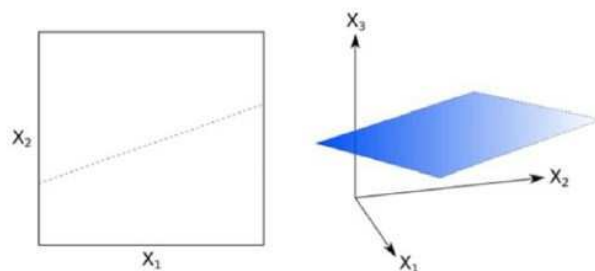


下图是以机器学习中的支持向量机(SVM)模型，来预测未来价格涨、跌的变化。SVM 可藉由创造非线性的 Kernel 函数，将特征空间调整到高维度的情况，对涨、跌的变化，进行非线性的预测。本例中使用 RBF Kernel 函数，透过 Scikit-Learn 套件，很容易便实作出预测的应用。

◇ 本課程介紹機器學習預測方法：使用 RBF Kernel 支持向量機預測價格漲跌

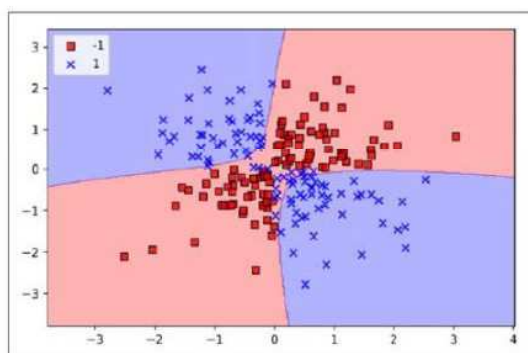
◆ 根據已知的樣本，將未知之物件分類。分類之原則，乃是根據這些物件之特徵(Features)。

➢ 特徵未必是直觀的物件屬性。例如，文件正、負情緒的分類。



◆ 使用 RBF Kernel 分割。

$$K(x^{(i)}, x^{(j)}) = \exp\left(-\frac{\|x^{(i)} - x^{(j)}\|^2}{2\sigma^2}\right)$$



```
In [2]: # Create the (parametrised) models
print("Hit Rates/Confusion Matrices:\n")
models = [("RSVM", SVC(C=1000000.0, cache_size=200, class_weight=None,
                        coef0=0.0, degree=3, gamma=0.0001, kernel='rbf',
                        max_iter=-1, probability=False, random_state=None,
                        shrinking=True, tol=0.001, verbose=False))]

# Train each of the models on the training set
for m in models:
    m[1].fit(X_train, y_train)
    pred = m[1].predict(X_test)
    print(m[0], m[1].score(X_test, y_test))
    print(confusion_matrix(pred, y_test))

Hit Rates/Confusion Matrices:

RSVM 0.5298804780876494
[[ 11  10]
 [108 122]]
```

本课程的影片都已经上网，課程大綱可參考連結如下

<https://github.com/andydong1209/PythonAIOnTrading/blob/main/AIOnTrading.pdf>

初级班《古典交易策略与机器学习交易策略》的课程链接：

<https://www.uquant.org/goods/show/94?targetId=75&preview=0>

中级班《时间数列交易策略》的课程链接：

<https://www.uquant.org/goods/show/97?targetId=78&preview=0>

高级班《深度神经网络交易策略》的课程链接：

<https://www.uquant.org/goods/show/98?targetId=79&preview=0>

有兴趣的朋友可以直接与我连络。

我的 WeChat ID: andydong3137。我的电邮: andydong1209@gmail.com。

请注明有兴趣 Python AI 金融交易策略实作班即可。