

ข้อสอบปลายภาค SC253 การโปรแกรมทางการเงิน

คำสั่ง กรณีเขียนโปรแกรมด้วยคอมพิวเตอร์

- ให้นักศึกษาเปลี่ยนชื่อ folder ที่ให้ download โดยตั้งชื่อเป็น เลขประจำตัวนักศึกษา เว้นวรรค ชื่อของนักศึกษา เช่น 2010511106050 พิชัย
- ให้ทำข้อสอบแต่ละข้อโดยแยกบันทึกลงในโฟลเดอร์ย่อยชื่อ Q1 ถึง Q5 ตามลำดับ
- เมื่อทำข้อสอบเสร็จให้ทำการบีบอัดทั้งโฟลเดอร์เป็นไฟล์ .zip เพียงไฟล์เดียว เช่นจากตัวอย่างด้านบนจะได้ไฟล์ชื่อว่า "2010511106050 พิชัย.zip" ให้ส่งไฟล์ดังกล่าวใน Assignment ชื่อว่า "Final Exam" ใน MS Teams

คำสั่ง กรณีเขียนคำตอบในกระดาษ (ไม่มีเครื่องคอมพิวเตอร์)

- หลังจากนักศึกษา download ข้อสอบ นักศึกษาสามารถดูรายละเอียดของข้อสอบแต่ละข้อและไฟล์ต่าง ๆ ที่ให้มาทาง iPad ในการตอบคำถามแต่ละข้อให้นักศึกษาเขียนหมายเลขข้อและชื่อไฟล์ที่เขียนให้ชัดเจน
- ถ่ายรูปคำตอบทั้งหมดส่งใน Assignment ชื่อว่า "Final Exam" ใน MS Teams

1. (6 คะแนน) จากระบบการซื้อขายหลักทรัพย์ต่อไปนี้ บางระบบอาจเกิดปัญหาการเกิดสัญญาณซื้อหลายครั้งติดต่อกันโดยไม่เกิดสัญญาณขาย และเกิดปัญหาการเกิดสัญญาณขายหลายครั้งติดต่อกันโดยไม่เกิดสัญญาณซื้อ จงอธิบายว่าระบบการซื้อขายหลักทรัพย์ต่อไปนี้เกิดหรือไม่เกิดปัญหาดังกล่าวเพราะเหตุใด

- ระบบดัชนีกำลังสัมพัทธ์ (relative strength index, RSI) ซึ่งเกิดสัญญาณซื้อเมื่อ RSI ตัด 30 ขึ้น และเกิดสัญญาณขายเมื่อ RSI ตัด 70 ลง
- ระบบการตัดกันของเส้นค่าเฉลี่ยเคลื่อนที่ (moving average crossover) ซึ่งเกิดสัญญาณซื้อเมื่อ EMA15 ตัด EMA75 ขึ้น และเกิดสัญญาณขายเมื่อ EMA15 ตัด EMA75 ลง
- ระบบแถบโบลลิงเจอร์ (Bollinger bands) ซึ่งเกิดสัญญาณซื้อเมื่อราคาปิดตัด lower band ขึ้น และเกิดสัญญาณขายเมื่อราคาปิดตัด upper band ลง

(ในข้อนี้ให้ตอบคำถามใน Word แล้วบันทึกเป็นไฟล์ชื่อ Q1.docx)

2. (8 คะแนน) จากไฟล์ KBANK.csv ที่ให้มาซึ่งเป็นไฟล์ราคาหลักทรัพย์รายวันของธนาคารกรุงไทย จำกัด มหาชน จงเขียนสคริปต์เพื่อ

- อ่านข้อมูลจากไฟล์ดังกล่าวไว้ใน data frame ชื่อ df
- จากนั้นคำนวณตัวบ่งชี้ (indicators) ต่อไปนี้

1. ค่าเฉลี่ยเคลื่อนที่อย่างง่าย 10 วันของราคาเปิด (SMA100)
2. ค่าเฉลี่ยเคลื่อนที่แบบ exponential 10 วันของราคาเปิด (EMA100)

- เพิ่มตัวบ่งชี้ทั้ง 2 ใน data frame df โดยใช้ชื่อที่อยู่ในวงเล็บเป็นชื่อคอลัมน์
- บันทึก data frame df ลงในไฟล์ชื่อ indicators.csv

(ในข้อนี้นอกเหนือจากไฟล์ KBANK.csv และไฟล์ที่อาจมีการ import ให้นักศึกษาส่ง 2 ไฟล์คือ ไฟล์ Q2.py ซึ่งเป็นสคริปต์ที่นักศึกษาเขียนขึ้นเองตามคำสั่งด้านบน และไฟล์ indicators.csv ซึ่งเป็นผลลัพธ์ หากนักศึกษาทำในกระดาษไม่ต้องส่งไฟล์ผลลัพธ์)

3. (8 คะแนน) จากฟังก์ชัน My_Sig ซึ่งยังไม่สมบูรณ์ต่อไปนี้ ซึ่งจะสร้างสัญญาณซื้อเมื่อเส้น RSI ตัด 30 ขึ้นและ MACD สูงกว่า 0 และสร้างสัญญาณขายเมื่อเส้น RSI ตัด 70 ลงและ MACD ต่ำกว่า 0 จงเขียนฟังก์ชันนี้ให้ทำงานได้สมบูรณ์ นักศึกษาอาจต้อง import module เพื่อคำนวณค่า RSI และ MACD หรือ อาจเขียนฟังก์ชันการคำนวณตัวบ่งชี้ทั้งสองขึ้นเอง พร้อมทั้งเขียนสคริปต์หลักเพื่อทดสอบการทำงานของฟังก์ชันนี้โดยมีการทำงานดังต่อไปนี้

- อ่านข้อมูลจากไฟล์ KBANK.csv ไว้ใน data frame ชื่อ df
- จากนั้นคำนวณตัวบ่งชี้ (indicators) ต่อไปนี้ โดยอาจเรียกใช้ฟังก์ชันจากการ import หรือ อาจเขียนขึ้นใหม่
 1. ค่าเฉลี่ยเคลื่อนที่ร่วมทาง/แยกทาง (MACD)
 2. ค่าดัชนีกำลังสัมพันธ์ซึ่งคำนวณโดยใช้ช่วงระยะเวลา 14 วัน (RSI14)
- เรียกฟังก์ชัน My_Sig เพื่อหาสัญญาณซื้อขายตามเงื่อนไขด้านบน
- เพิ่มตัวบ่งชี้ทั้ง 2 และสัญญาณซื้อขายใน data frame df (เพิ่ม 3 คอลัมน์)
- บันทึก data frame df ลงในไฟล์ชื่อ My_Signal.csv

```
def My_Sig(RSI14,MACD) :
    rows = len(RSI14)
    signal = [0]*rows
    for i in range(1,rows) :
        ...
        ...
    return signal
```

(ในข้อนี้นอกเหนือจากไฟล์ KBANK.csv และไฟล์ที่อาจมีการ import ให้นักศึกษาส่ง 2 ไฟล์คือ ไฟล์ Q3.py ซึ่งเขียนฟังก์ชัน My_Sig พร้อมทั้งสคริปต์หลัก และไฟล์ผลลัพธ์ My_Signal.csv หากนักศึกษาทำในกระดาษไม่ต้องส่งไฟล์ผลลัพธ์)

4. (8 คะแนน) จากไฟล์ trade.py ที่ให้มา จงแก้ไขฟังก์ชัน tradeSim ตามรายละเอียดด้านล่าง

```
def tradeSim(df, signal, filename):
    Open = df['Open']
    rows = len(Open)
    num = 0
    total = 0
    Cost = 0
    numBuy = [np.nan]*rows
    cost = [np.nan]*rows
    numSell = [np.nan]*rows
    income = [np.nan]*rows
    profit = [np.nan]*rows
    ## Buy & Sell Simulation
    for i in range(1, rows-1):
        if (signal[i-1] == -1) and (signal[i] == 1):
            numBuy[i+1] = floor((100000/(Open[i+1]*1.0017))/100)*100
            num = numBuy[i+1]
            cost[i+1] = -numBuy[i+1]*Open[i+1]*1.0017
            Cost = cost[i+1]
        elif (signal[i-1] == 1) and (signal[i] == -1):
            numSell[i+1] = -num
            income[i+1] = -numSell[i+1]*Open[i+1]*0.9983
            profit[i+1] = income[i+1]+Cost;
            total = total + profit[i+1]
    filename = 'tradesim/' + filename
    df['Signal'] = signal
    df['NumBuy'] = numBuy
    df['Cost'] = cost
    df['NumSell'] = numSell
    df['Income'] = income
    df['Profit'] = profit
    df.to_csv(filename)
    total = total/100000/3*100
    return total
```

ในฟังก์ชัน tradeSim ให้มีการเปลี่ยนแปลงรายละเอียดต่าง ๆ ดังต่อไปนี้

- แต่ละรอบการซื้อขายมีงบประมาณไม่เกิน 50,000 บาท
- ลดอัตราค่าคอมมิชชั่นและค่าธรรมเนียมทั้งในการซื้อและการขายจาก 0.17% เป็น 0.10%
- เปลี่ยนขนาดหน่วยการซื้อขาย (board lot) จาก 100 หุ้นเป็น 50 หุ้น
- ใช้ทดสอบกับข้อมูลราคาหลักทรัพย์ในช่วงระยะเวลา 5 ปี แทนช่วงระยะเวลา 3 ปี

และทดสอบการทำงานของฟังก์ชันดังกล่าวโดยการ run ไฟล์ชื่อ Q4.py

(ในข้อนี้ให้ส่งทั้งไฟล์เดือรที่ให้มาโดยให้ทำการแก้ไขฟังก์ชัน tradeSim ในไฟล์ trade.py เท่านั้น)

5. (10 คะแนน) จากฟังก์ชัน My_Sig ในไฟล์ My_Signal.py ซึ่งจะสร้างสัญญาณซื้อเมื่อเส้น RSI (ซึ่งคำนวณโดยใช้ระยะเวลา 14 วัน) ตัด 30 ขึ้น และสร้างสัญญาณขายเมื่อเส้นค่าเฉลี่ยเคลื่อนที่ระยะสั้น (short) ตัดเส้นค่าเฉลี่ยเคลื่อนที่ระยะยาว (long) ลง หากต้องการทดสอบพารามิเตอร์ต่าง ๆ ดังต่อไปนี้

- ระยะเวลาในการคำนวณ RSI (rsiPeriods) เป็น 4, 9, 14 จากเดิมที่กำหนดเป็น 14 เพียงค่าเดียว
- ระยะเวลาที่ใช้ในการคำนวณค่าเฉลี่ยเคลื่อนที่ระยะสั้น (shortPeriods) เป็น 5, 10, 15, 20 จากเดิมที่กำหนดเป็น 10 เพียงค่าเดียว
- ระยะเวลาที่ใช้ในการคำนวณค่าเฉลี่ยเคลื่อนที่ระยะยาว (longPeriods) เป็น 25, 50, 75, 100 จากเดิมที่กำหนดเป็น 50 เพียงค่าเดียว

จงแก้ไขโค้ดในฟังก์ชัน My_Sig เพื่อให้สามารถทดสอบพารามิเตอร์ต่าง ๆ ดังที่กำหนดได้พร้อมทั้งเขียนสคริปต์เพื่อทดสอบพารามิเตอร์เหล่านั้นและบันทึกในชื่อ Q5.py

```
from RSI import rsi
from EMACO import emaco

def My_Sig(close):
    rows = len(close)
    signal = [0] * rows
    [short, long] = emaco(close, 10, 50)
    RSI = rsi(close, 14)
    for i in range(1, rows):
        if (RSI[i-1]<30) and (RSI[i]>30):
            signal[i] = 1
        elif short[i-1]>=long[i-1] and short[i]<long[i]:
            signal[i] = -1
    return signal
```

(ในข้อนี้ให้ส่งทั้งไฟล์เดอร์ที่นำมาโดยให้ทำการแก้ไขฟังก์ชัน My_Sig ในไฟล์ My_Signal.py และเขียนสคริปต์หลักเพื่อทดสอบพารามิเตอร์ตามรายละเอียดด้านบนและบันทึกในชื่อ Q5.py)