



מעבדה לניתוח נתונים פיננסיים - עבודה מסכמת

הגשה: יום ו' 15.03.2024 בשלשות (לפחות סטודנט אחד מכלכלה ואחד ממדמ"ח בכל שלשה)

הנחיות ודגשים כלליים:

1. קראו את כל העבודה כולל הנספחים לפני שאתם מתחילים לעבוד.
2. יש להגיש את העבודה בקובץ zip אחד שמכיל קובץ וורד אחד עם התשובות והתוצאות של כל הסעיפים ואת שאר קבצי הקוד כמפורט בכל סעיף.
3. יש לרשום את שמות ות"ז המגישים והמגישות בתחילת קובץ הוורד.
4. על כל חברי וחברות הצוות להכיר ולהבין כל חלק בעבודה ללא יוצא מן הכלל.
5. על כל חברי וחברות הצוות לפתוח משתמש ב-WRDS.
6. על כל חברי וחברות הצוות להריץ את הקוד בחלק 1 במחברת WRDS במשתמש האישי ואת שאר הקוד במחשב האישי.
7. הוורד צריך להיות נקי ומסודר עם כתב וגודל אחידים (למעט כותרות).
8. הקוד צריך להיות נקי, מסודר ולהסביר את עצמו (לא להוסיף תיעוד בכל מקום) וללא magic strings-I magic numbers.
9. אין להשתמש ב-GPT לביצוע המשימות במלואן ולכתיבת התשובות, כן מותר ואף מומלץ להיעזר ב-GPT בכתיבת הקוד.
10. יש להגיש את העבודה בעברית.
11. יתקיימו מפגשי זום ביחידים ובקבוצות לבחינה פרונטלית של העבודה.

חלק 1 – ניקוי ועיבוד נתונים ב-WRDS

1. פתחו מחברת עבודה ב-WRDS.
2. העלו את הקובץ final_task_bad_data.pickle לסביבת העבודה של המחברת.
3. טענו את הקובץ ל-pandas DataFrame.
4. תארו מה הן הבעיות שיש בנתונים בפירוט לפי עמודות.
5. כיצד אתם מציעים להשלים ולתקן את הנתונים, בהנחה שמשימתנו הצפויה היא ניתוח סטטיסטי היסטורי של המניה? פרטו למה בחרתם בשיטה זו לכל אחת מהעמודות בנפרד.
6. כתבו קוד לתיקון הנתונים בדרכים שהצעתם ושמרו את הנתונים המתוקנים בקובץ pickle בשם fixed_data.pickle.
7. האם הפתרון שהצעתם חסין בפני data leakage כפי שהגדרנו אותו בכיתה? אם כן הסבירו למה כן, ואם לא הסבירו למה לא ואיך הייתם פותרים זאת, והציעו פתרון אחד (ללא צורך לממש בקוד).
8. כעת טענו מ-CRSP את הנתונים היומיים (ohlcv, dividends, splits_adjustment_factor) של מניית Abercrombie & Fitch Co. מהראשון לינואר 2018 עד הראשון לאפריל 2020.
9. חשבו את ה-adjusted close של המניה בהתאם לספליטים והדיבידנדים שלה.
10. בדקו מול ה-historical data ב-yahoo finance אם החישוב שלכם נכון (מומלץ לבדוק מול תאריכים ישנים שחולקו אחריהם לפחות שני דיבידנדים).
11. חשבו את ה-adjusted open, adjusted high, adjusted low ו-adjusted close בהתאם ל-adjusted close.
12. שמרו את המחברת בשם part1.ipynb והורידו אותה למחשב שלכם.



חלק 2 – ניתוח טכני

1. צפו בסרטון על איך מתעשרים מניתוח טכני בקישור: [התעשרות בקליק](#).
2. הורידו את הנתונים של BTCUSDT מבורסת Binance באמצעות הקוד שמימשנו בכיתה (שימו לב שאלו הנתונים של BTCUSDT.P ב-trading view) מהראשון לינואר 2023 עד הראשון לינואר 2024 באינטרוולים של חצי שעה.
3. חשבו את כל האינדיקטורים הטכניים בהם המציג משתמש בסרטון וכתבו את הנוסחה המתמטית של אחד מהם. הסבירו בקצרה מה ההיגיון שעומד מאחוריו (למשל, ההיגיון של BB הוא להראות איפה המחיר נמצא ביחס לסטיות התקן שלו ובכך כמה המחיר הנוכחי חריג סטטיסטית).
4. בדקו מול האינדיקטורים ב-trading view שהחישובים שלכם מדויקים.
5. תארו את אסטרטגיית המסחר מבחינה לוגית ואת ההיגיון שעומד מאחוריה. כתבו האם אתם מאמינים בהיגיון זה או לא ולמה, וכתוצאה מכך העריכו האם האסטרטגיה תרוויח או תפסיד.
6. ממשו את האסטרטגיה של המציג ומדדו את הביצועים שלה באמצעות הכלים שלמדנו. פרטו את התוצאות שקיבלתם.
7. המציג המליץ להשתמש בפרמטרים שונים לחישוב האסטרטגיה. בחרו פרמטר אחד, הסבירו על מה הוא משפיע באסטרטגיה וכיצד השינוי שלו ישפיע על התנהגות האינדיקטור. הציגו עבורו ערך חלופי וחזרו על סעיף 5 עם השינוי של פרמטר זה.
8. עבור סעיפים 5 ו-6, הציגו בגרף את פעולות האסטרטגיה לאורך התקופה, ובגרף נוסף הציגו ניתוח סטטיסטי של הרווחים שלה (אחוזי השינוי ב-portfolio value).
9. השוו את ההערכה שלכם לתוצאות שקיבלתם, האם הן תואמות? הסבירו.
10. רשמו את התוצאות והגרפים שלכם בוורד ושמרו את הקוד שכתבתם בשם part2.ipynb או part2.py, לפי האופן בו כתבתם אותו (אם כתבתם אותו כפרויקט, אגדו את כל קבצי הפרויקט ב-part2.zip).

חלק 3 – מציאת ארביטראז'

שימו לב שמכפיל נכס הבסיס בכל אופציה הוא 100, ומחיר כל אופציה וכן מחיר המימוש נקובים עבור רכישה/מכירה של יחידה אחת של נכס הבסיס. לכן חישוביכם צריכים להיות בכפולות של 100 גם עבור מחיר האופציה, גם עבור מחיר המימוש וגם עבור מחיר נכס הבסיס.

1. טענו את הנתונים של האופציות על חוזה QQQ בשנת 2023.
2. נקו שורות בהן נפח המסחר קטן מ-100.
3. מצאו את העסקה הרווחית ביותר בכל יום המבוססת על ארביטראז' PCP בהנחה שניתן לבצע עסקה אחת ביום הכוללת 100 יחידות מנכס הבסיס וקניה או מכירה של אופציה אחת מכל סוג.
4. כעת חשבו מה הוא הרווח השנתי אותו הייתם מרוויחים מביצוע עסקאות אלה ורשמו את הפעולות שעליכם היה לבצע בחמשת הימים הכי רווחיים על מנת לעשות זאת.
5. רשמו את התוצאות והפעולות שלכם בוורד ושמרו את הקוד שכתבתם בשם part3.ipynb או part3.py לפי האופן בו כתבתם אותו (אם כתבתם אותו כפרויקט, אגדו את כל קבצי הפרויקט ב-part3.zip).



UT Bot pseudo code – 'א' נספח

```
def TR(high, low, prev_close):
    return max(high, prev_close) - min(low, prev_close)

def ATR(tr, atr_length):
    atr = [0...0] # init
    atr[atr_length - 1] = mean(tr[0], tr[1],...,tr[atr_length -1])
    for i from atr_length to num_rows:
        atr[i] = (atr[i - 1] * (atr_length - 1) + tr[i]) / atr_length
    return atr

def crossover(a, b):
    return curr_a > curr_b and prev_a < prev_b

def UTBot(data, key_value, atr_length):
    tr = TR(data.high, data.low, data.prev_close)
    atr = ATR(tr, atr_length)

    loss_threshold = key_value * atr

    trailing_stop = [0, ..., 0] # init

    for i from 1 to num_rows:
        if close[i] > trailing_stop [i-1] and close[i-1] > trailing_stop [i-1]:
            trailing_stop [i] = max(trailing_stop, close[i] - loss_threshold[i])
        elif close[i] < trailing_stop [i-1] and close[i-1] < trailing_stop [i-1]:
            trailing_stop [i] = min(trailing_stop [i-1], close[i] + loss_threshold[i])
        elif close[i] > trailing_stop[i-1]:
            trailing_stop[i] = close[i] - loss_threshold.iloc[i]
        else:
            trailing_stop[i] = close[i] + loss_threshold.iloc[i]

    above = crossover(close, trailing_stop) # close cross trailing_stop up
    below = crossover(trailing_stop, close) # close cross trailing_stop down

    buy_signals = int(above)
    sell_signals = int(below)
```



נספח ב' – STC Oscillator pseudo code

```
def MacdDiff(close, fast_length, slow_length):
    fast_ema = ema(close, fast_length)
    slow_ema = ema(close, slow_length)
    return fast_ema - slow_ema

def SmoothSrs(srs, smoothing_f):
    smoothed_srs = [0, ..., 0] # init
    smoothed_srs[0] = srs[0]

    for i from 1 to num_rows:
        if smoothed_srs[i-1] == Na:
            smoothed_srs[i] = srs[i]
        else:
            smoothed_srs[i] =
                smoothed_srs[i-1] + smoothing_f * (srs[i] - smoothed_srs[i-1])
    return smoothed_srs

def NormalizeSmoothSrs(series, window_length, smoothing_f):
    lowest = rolling_min(series, window_length)
    highest_range = rolling_max(series, window_length) - lowest

    normalized_series = (element-wise)
        if highest_range > 0:
            normalized_value = (series - lowest) / highest_range * 100
        else:
            normalized_value = Na
    Forward_fill(normalized_series)

    smoothed_series = SmoothSrs(normalized_series, smoothing_f)
    return smoothed_series

def STC(close, stc_length, fast_length, slow_length, smoothing_factor=0.5):
    macd_diff = MacdDiff(close, fast_length, slow_length)
    normalized_macd = NormalizeSmoothSrs(macd_diff, stc_length, smoothing_factor)
    final_stc = NormalizeSmoothSrs(normalized_macd, stc_length, smoothing_factor)
    return final_stc
```