

## Take Home Midterm Exam (Individual)

ITE351

**DUE: Check LMS for online submission**

(This is not an easy exam for sure) This is an **individual “Take Home Midterm Exam”** You are asked to complete the following tasks. For this exam, **DO NOT** use Kakao Talk asking questions to peers. If you have questions, please send me an email (youngjoon@hanyang.ac.kr). If it's a feasible question, I will reply back to you ASAP. If not, I will not get back to you. Consider it like an open book exam.

**READ INSTRUCTIONS VERY CAREFULLY!**

You are doing ML analysis for trading cryptocurrency to make a profit. Our target cryptocurrency is one of popular ones called BitCoin (BTC). KRW below means Korean Won.

**\*2019-05-trade.csv** data format: This file contains the trading history of BTC-KRW market during May 16-31, 2019. It's a basically a file showing the sequence of Buy and Sell made for BTC. And it may be possible to use this data as training/test dataset for smart transaction agent for cryptocurrency.

- timestamp: time of event in place, yyyy-mm-dd HH:MM
- quantity: BTC size in trade
- price: 1 BTC price in KRW
- fee: ignore this
- amount: the total amount of KRW in trade, quantity \* price
- side: 0 means Buy (also known as 'Bid'), 1 means Sell (also known as 'Ask')

Here is one example from May 16, 2019.

...

2019-05-16 13:40, 0.20, 9549000, 954.90, 1908845, 1

...

At May-16-2019 13:40, 0.20 BTC was sold (side: 1) at the price of 9,549,000 KRW.

**\*2019-05-17-BTC-orderbook.csv** data format: This file contains so-called orderbook of BTC market. The data has the historical records of willingness to Buy and Sell BTC for every second. Every second data have multiple lines of Buy and Sell. The first several lines represent Buy requests, the next several lines represent Sell requests.

- price: 1 BTC price in KRW
- quantity: BTC size is willing to Buy or Sell.
- type: 0 means Buy ('Bid'), 1 means Sell ('Ask')
- timestamp: time of market, yyyy-mm-dd HH:MM:SS.us

the following (orderbook data) example is from May 17, 2019.

...

9435000.0, 1.6979, 0, 2019-05-17 00:00:00.962338	← Top Level Buy (top_bid_price)
9434000.0, 0.0015, 0, 2019-05-17 00:00:00.962338	← Level 2 Buy
9433000.0, 0.0018, 0, 2019-05-17 00:00:00.962338	← Level 3 Buy
9431000.0, 0.0475, 0, 2019-05-17 00:00:00.962338	
9430000.0, 0.8173, 0, 2019-05-17 00:00:00.962338	
9450000.0, 24.1714, 1, 2019-05-17 00:00:00.962338	← Top Level Sell (top_ask_price)
9455000.0, 0.2023, 1, 2019-05-17 00:00:00.962338	← Level 2 Sell
9458000.0, 0.0112, 1, 2019-05-17 00:00:00.962338	← Level 3 Sell
9459000.0, 0.3099, 1, 2019-05-17 00:00:00.962338	
9462000.0, 1.3064, 1, 2019-05-17 00:00:00.962338	

...

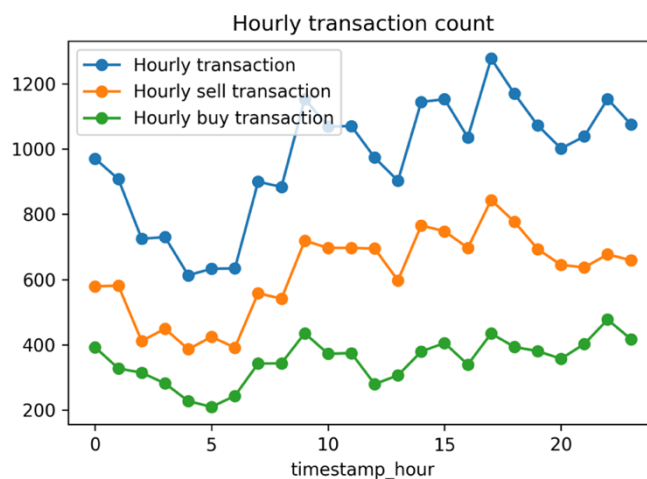
At Top Level Buy, someone wants to buy 1.6979 BTC at the price of 9,435,000 KRW in May-17-2019 00:00:00 (right at the midnight). At Top Level Sell, someone wants to sell 24.1714 BTC at the price of 9,450,000 KRW. So these datasets show the buy and sell requests in the market. Somebody can make some money!

**\*Task 1:** A file you need to complete Task 1: 2019-05-trade.csv. Compute the total profit of May in KRW. It simply means how much money do we make or lose? To calculate the exact profit over the days, you should be calculating when the accumulative quantity is close to 0. The accumulative quantity infers the moment that the number of quantities bought and sold are equal (the difference between them is close to 0). Only consider 4-digit floating number, ignore the rest when you are calculating. Show the difference between how much KRW we spent to buy and sell at that moment(s). This is the “exact profit” of the 2019-05-trade.

Or there is a simpler way to calculate “approximate profit” using ‘amount’ column. You can easily figure this one out. If your answer is to show “exact profit” and its process, a full mark (100%) will be given for task 1. If your answer is to show “approximate profit”, an 80% will be given.

Show me your codes. If you like to explain your steps, please do so. Write them down clearly or your concerns.

**\*Task 2:** A file you need to complete Task 2: 2019-05-trade.csv. Report how many Buy and Sell trades separately. Draw a daily time-series bar graph illustrating changes in transaction counts (x-axis: days, y1-axis: Sell, y2-axis: Buy). The following sample graph is drawn hourly. Your graphs would look similar but in days.



**\*Task 3:** Files you need to complete Task 2: 2019-05-trade.csv and 2019-05-17-BTC-orderbook.csv. Compute the following features and modify 2019-05-trade.csv. Show the first 20 and last 20 lines of your new csv data. For the new csv data, you will remove a few existing columns and add three new columns: MidPrice, Bfeature, and Alpha. In order to compute these three columns, check out the following:

\*How to compute MidPrice: ask means Sell, bid means Buy.

$$\text{MidPrice} = (\text{top\_ask\_price} + \text{top\_bid\_price}) / 2$$

\*How to compute Bfeature

```
askQty = orderbook_ask_quantity.avgerage() # average quantity of all levels for Sell (side 1)
bidQty = orderbook_bid_quantity.avgerage() # likewise for Buy (side 0)
bidPx = orderbook_bid_price.avgerage()      # average price of all levels for Buy (side 0)
```

```
book_price = (askQty*bidPx)/bidQty
Bfeature = (book_price - mid_price)
```

\*How to compute Alpha:

$$\text{Alpha} = \text{Bfeature} * \text{MidPrice}$$

\* Your new 2019-05-trade.csv data format will be:

**timestamp, quantity, price, midprice, bfeature, alpha, side**

For example, the first line of 2019-05-17 in 2019-05-trade.csv:

2019-05-17 00:00, 0.05770069, 9449000, 272.60, 544941, 1

It happened at 2019-05-17 00:00. So use the very first sec dataset at 00:00:00.962338 from 2019-05-17-BTC-orderbook.csv. Use the following dataset to compute **midprice, bfeature, alpha** for 2019-05-17 00:00 and add them in column to your new 2019-05-trade.csv file.

```
9449000.0, 40.39262446, 0, 2019-05-17 00:00:00.962338
9448000.0, 0.17676982, 0, 2019-05-17 00:00:00.962338
9446000.0, 2.11887902, 0, 2019-05-17 00:00:00.962338
9445000.0, 0.02805574, 0, 2019-05-17 00:00:00.962338
9441000.0, 0.74081055, 0, 2019-05-17 00:00:00.962338
9440000.0, 0.2338522, 0, 2019-05-17 00:00:00.962338
9439000.0, 0.01637451, 0, 2019-05-17 00:00:00.962338
9438000.0, 0.29291399, 0, 2019-05-17 00:00:00.962338
9437000.0, 0.48892999, 0, 2019-05-17 00:00:00.962338
9436000.0, 0.11283857, 0, 2019-05-17 00:00:00.962338
9435000.0, 1.69795923, 0, 2019-05-17 00:00:00.962338
9434000.0, 0.00155999, 0, 2019-05-17 00:00:00.962338
9433000.0, 0.00189399, 0, 2019-05-17 00:00:00.962338
9431000.0, 0.04756431, 0, 2019-05-17 00:00:00.962338
9430000.0, 0.81736619, 0, 2019-05-17 00:00:00.962338
9450000.0, 24.17141283, 1, 2019-05-17 00:00:00.962338
9455000.0, 0.20238048, 1, 2019-05-17 00:00:00.962338
9458000.0, 0.01123548, 1, 2019-05-17 00:00:00.962338
9459000.0, 0.30999, 1, 2019-05-17 00:00:00.962338
9462000.0, 1.30642, 1, 2019-05-17 00:00:00.962338
```

```
9465000.0, 0.44874552, 1, 2019-05-17 00:00:00.962338
9466000.0, 1.0, 1, 2019-05-17 00:00:00.962338
9473000.0, 0.31419, 1, 2019-05-17 00:00:00.962338
9475000.0, 3.061, 1, 2019-05-17 00:00:00.962338
9480000.0, 1.29058544, 1, 2019-05-17 00:00:00.962338
9481000.0, 0.29193, 1, 2019-05-17 00:00:00.962338
9482000.0, 0.0517347, 1, 2019-05-17 00:00:00.962338
9483000.0, 0.488, 1, 2019-05-17 00:00:00.962338
9490000.0, 0.08680052, 1, 2019-05-17 00:00:00.962338
9491000.0, 0.699, 1, 2019-05-17 00:00:00.962338
```

Search for the next trade row, 2019-05-17 00:02, and find the 2019-05-17 00:02:00 in the orderbook data file. And compute each feature again. You are repeating this for every timestamp in 2019-05-17.

A full mark will be given if you find the corresponding timestamps in the trade and orderbook files and add the midprice, bfeature, and alpha to your new trade data. Only for the timestamps that exists in the trade file! Some timestamps are missing in the orderbook data file, then just omit them (Just fill 0).

**\*(BONUS) Task 4:** How can you use the data file from Task 3 to create the smart trading agent? You can show how to create train and test datasets. And show how to use ML (using R or python random-forest exercise from our class or others like PCA, anything that you like) to create the learning agent for cryptocurrency transaction. What's going to be an appropriate target feature here and what about the remaining features? If you can provide a code and running example (by attaching the screenshot), that will be the best. Explanation in words is also okay. Be clear and show code if you can (If you want, you can explain in the code comment lines). Accuracy does not matter. Take a good look at the samples that we have seen in the class, perhaps use them wisely.

For example,

- show how to manipulate the dataset (reading, processing, etc.)
- show how to split training/test dataset
- show how to train or make the model
- show how to use the built model to test

#### Submission:

Submit a single PDF file having Taks-1, 2, 3, and 4 in order. In your PDF file, please include your source code (if any), figures, and screenshots of running code (if any). (notebook to pdf export would be the best)

Do as much as you can and submit. **It's normal that you cannot finish the whole exam.**

#### Suggestions:

Take a very similar step like the examples we saw in class. You will show me how to investigate the dataset. You can choose one of the methods from the AI/Machine learning/DL libraries (e.g., <https://scikit-learn.org/stable/>, R, Weka <https://www.cs.waikato.ac.nz/ml/weka/>). Pandas should be really helpful. You are showing a step-by-step procedure on how to solve each task. It's completely up to you for your choice of tools.