Take Home Midterm Exam (Individual)

ITE351

DUE: Check LMS for online submission

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 direct to me via 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.

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 EOS. KRW means Korean Won.

*2018-06-trade.csv data format: This file contains the trading history of EOS-KRW market during June 1-2, 2018. It's a basically a file showing the sequence of Buy and Sell made for EOS. And it may be possible to use this data as training/test dataset for smart transaction agent for cryptocurrency.

- timestamp:	time of your order in place, yyyy-mm-dd HH:MM:SS
- quantity:	EOS size in transaction
- price:	1 EOS price in KRW (Korean Won)
- fee:	ignore this
- amount:	the total amount of KRW in trade, quantity * price
- side:	0 means Buy (Bid), 1 means Sell (Ask)

The following example is from June 2018.

```
…

2018-06-01 00:01:34, 22.075, 13590, 0, 299999, 1
…
```

At June-1-2018 01:01:34, we sold 22.075 EOS at the price of 13,590 KRW.

*2018-06-01-orderbook.csv and 2018-06-02-orderbook.csv data format: This file contains so-called orderbook of EOS-KRW market. The data has the history records of willingness to Buy and Sell EOS for every second. Every second data have multiple levels of Buy and Sell. The first several lines represent Buy requests, the next several lines represent Sell requests. This happens every second.

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

Again, the following (orderbook) example is from June, 2018.

…

```
13520, 6457.2842, 0, 2018-06-01 00:00:00.155449
13510, 2453.7794000000004, 0,2018-06-01 00:00:00.155449
13500, 9035.092100000002, 0, 2018-06-01 00:00:00.155449   ← Level 3 Buy
13480, 2963.2203999999997, 0, 2018-06-01 00:00:00.155449  ← Level 2 Buy
13470, 1216.4360000000001, 0, 2018-06-01 00:00:00.155449  ← Top Level Buy (top_bid_price)
13530, 163.8964, 1, 2018-06-01 00:00:00.155449            ← Top Level Sell (top_ask_price)
13540, 1014.0962, 1, 2018-06-01 00:00:00.155449           ← Level 2 Sell
13550, 14843.010063470001, 1, 2018-06-01 00:00:00.155449  ← Level 3 Sell
13570, 357.7872, 1, 2018-06-01 00:00:00.155449
13580, 2470.1464175300002, 1, 2018-06-01 00:00:00.155449
13590, 2582.4676, 1, 2018-06-01 00:00:00.155449
```

…

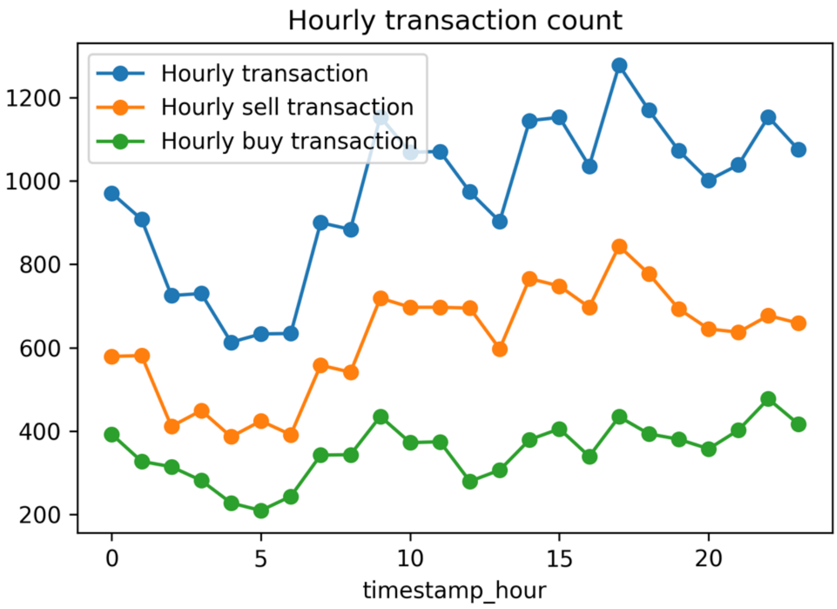
At Top Level Buy, someone wants to buy 1216.436 EOS at the price of 13,470 KRW in June-1-2018 00:00:00 (at the midnight). At Top Level Sell, someone wants to sell 163.8964 EOS at the price of 13,530 KRW. So these datasets show the buy and sell requests in the market. Somebody can make some money!

*Task 1: File(s) you need to complete Task 1: 2018-06-trade.csv. Compute the total profit of June 1-2 in KRW. It simply means how much money do we make or lose? To calculate the exact profit over two 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 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 2018-06-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 will be given for task 1. If your answer is to show “approximate profit”, a half 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: File(s) you need to complete Task 2: 2018-06-trade.csv. Report how many Buy and Sell trades separately. Draw a time-series bar graph illustrating changes in transaction counts (x-axis: 1 hour interval, y1-axis: Sell, y2-axis: Buy) The sample graph should be similar (it’s not the same as your answer) to the following figure.



*Task 3: File(s) you need to complete Task 2: 2018-06-trade.csv, 2018-06-01-orderbook.csv, 2018-06-02-orderbook.csv. Compute the following features and modify 2018-06-trade.csv. Copy and past your new csv data into PDF. For your 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.

MidPrice = (top_ask_price + top_bid_price) / 2

*How to compute Bfeature

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

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

*How to compute Alpha:

Alpha = Bfeature * MidPrice * 0.004

* Your new 2018-06-trade.csv data format now:

timestamp, price, midprice, bfeature, alpha, side

A full mark will be given if you find the corresponding timestamps in the orderbook and trade 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 trade data, then use the first corresponding timestamp (first second of every minute).

A half will be given if you calculate these three values for all timestamps in 06-01-orderbook (if you do this way, copy and past only first 120 seconds data only. Printing timestamp and three features is just fine).

*Task 4: (Bonus) Explain your plan how you use the data file from Task 3 to create the smart trading agent. You can explain how you create training and test dataset. And show how to use ML (using random-forest exercise from our class or others like PCA, anything that you like) to create the learning agent for cryptocurrency transaction. If you can provide a code and running example (by attaching the screenshot), that will be the best. If you think that other decision making (statistic-based) algorithms are suitable, then do so and explain how it worked. Be clear and show code if you can. 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 create training/test dataset
- show how to train or make the model
- show how to fit/accuracy check for prediction

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)

We will discuss more in class.

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

Some 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 - https://cran.r-project.org/web/views/MachineLearning.html, 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 do learning. Accuracy of your method DOES NOT matter in this exam.

It's completely up to you for your choice of tools.