Take Home Exam: Project-Assignment

AI+X: Deep Learning

DUE: June 12 *Submit in Class* (Possible deadline extension)

This is an individual “Take Home 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 feasible, I will reply back to you as soon as possible.

You are doing ML/DL analysis for trading cryptocurrency to make a profit. Our target cryptocurrency is popular Bitcoin (BTC).

*2018-05-trade.csv data format: This file contains the transaction history records of BTC-KRW market during May 1-2, 2018. It’s a basically a file showing the sequence of Buy and Sell transactions for BTC. And it may be possible to use this data as training/test dataset for smart transaction agent for cryptocurrency market.

- timestamp:	time of your order in place, yyyy-mm-dd HH:MM:SS
- quantity:	BTC size in transaction
- price:	1 BTC price in Korean Won (KRW)
- fee:	commission, you can ignore this
- amount:	the total amount of KRW in transaction, quantity x price
- side:	0 → Buy (Bid), 1 → Sell (Ask)

For example,

```
…
2018-05-01 01:06:24,0.3382,10162000,0,3436788,1
…
```

At May-1-2018 01:06:24, I sold 0.3382 BTC for the price of 10,162,000 KRW.

*2018-05-01-orderbook.csv and 2018-05-02-orderbook.csv data format: This file contains so-called orderbook (market data) of BTC-KRW market. The data has the order records of willingness to Buy and Sell BTC for every second. Every second data have top 15 levels of Buy and Sell (the first 10 lines are representing Buy requests, the next 15 lines are representing Sell requests, so the total of 30 lines are recorded for every second.)

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

For example,

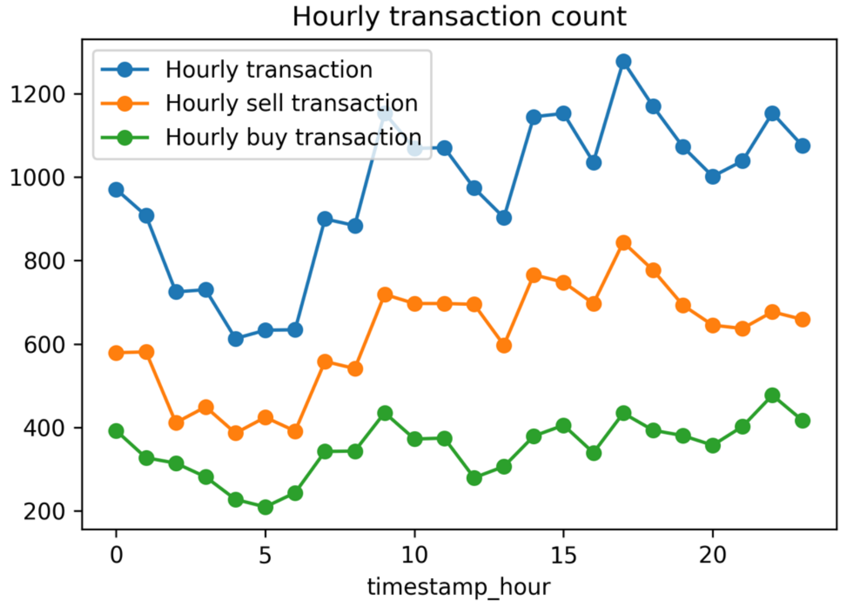
```
…
10118000,0.528,0,2018-05-01 00:00:01.491007
10115000,0.1018,0,2018-05-01 00:00:01.491007
10110000,0.1648,0,2018-05-01 00:00:01.491007
10109000,0.2331,0,2018-05-01 00:00:01.491007
10102000,1.0,0,2018-05-01 00:00:01.491007
10101000,8.0,0,2018-05-01 00:00:01.491007
10100000,0.322,0,2018-05-01 00:00:01.491007
10098000,0.1645,0,2018-05-01 00:00:01.491007
```

```
10094000,0.0335,0,2018-05-01 00:00:01.491007
10092000,0.1,0,2018-05-01 00:00:01.491007
10091000,0.0699,0,2018-05-01 00:00:01.491007
10086000,0.9911,0,2018-05-01 00:00:01.491007
10085000,0.816,0,2018-05-01 00:00:01.491007          ← Level 3 Buy
10083000,0.1194,0,2018-05-01 00:00:01.491007         ← Level 2 Buy
10082000,0.3807,0,2018-05-01 00:00:01.491007         ← Top Level Buy (top_bid_price)
10120000,0.016,1,2018-05-01 00:00:01.491007          ← Top Level Sell (top_ask_price)
10128000,0.4525,1,2018-05-01 00:00:01.491007         ← Level 2 Sell
10129000,0.384,1,2018-05-01 00:00:01.491007          ← Level 3 Sell
10130000,0.0011,1,2018-05-01 00:00:01.491007
10133000,1.04,1,2018-05-01 00:00:01.491007
10137000,0.6674,1,2018-05-01 00:00:01.491007
10138000,0.052,1,2018-05-01 00:00:01.491007
10139000,1.51777557,1,2018-05-01 00:00:01.491007
10140000,0.67156629,1,2018-05-01 00:00:01.491007
10143000,0.1669,1,2018-05-01 00:00:01.491007
10144000,1.3543,1,2018-05-01 00:00:01.491007
10145000,2.4118,1,2018-05-01 00:00:01.491007
10148000,0.1069,1,2018-05-01 00:00:01.491007
10149000,2.8576,1,2018-05-01 00:00:01.491007
10150000,0.7716,1,2018-05-01 00:00:01.491007
…
```

At top level Buy, someone wants to buy 0.3807 BTC at the price of 10,082,000 KRW in May-1-2018 00:00:01 (1 second after midnight). At top level Sell, someone wants to sell 0.016 BTC at the price of 10,120,000 KRW. So these datasets show the requested buy and sell orders in the market. Somebody can make some money!

*Task 1: Compute the total profit of May 1-2 in KRW. It simply means how much money do we make or lose? The profit calculation moment should be when the accumulative quantity is 0 (only consider 4 digit floating number, ignore the rest).

*Task 2: Report how many Buy and Sell transactions separately. Draw a time-series graph illustrating changes in transaction counts (x-axis: hour, y1-axis: Sell, y2-axis: Buy, y3-axis: Sell+Buy). The sample graph should be similar to the following figure.



*Task 3: For this task, you will need orderbook.csv files to work with. Compute the following features and modify 2018-05-trade.csv. Submit your new csv file. For your final csv file, you will remove a few existing columns and add two new columns: midprice and bookfeature. In order to compute these two, 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 BookFeature

```
askQty = quant_orderbook_ask.values.avg()      # average quantity of all levels for Sell
bidQty = quant_orderbook_bid.values.avg()      # likewise for Buy
askPx = price_orderbook_ask.values.avg()       # average price of all levels for Sell
bidPx = price_orderbook_bid.values.avg()       # likewise for Buy
```

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

* Your new 2018-05-trade.csv data format now:

timestamp, price, midprice, bookfeature, side

You have two the match the timestamps from orderbook.csv to the corresponding entries in 2018-05-trade.csv file.

if you have any additional feature that you come up with, please explain and include it to csv. I can give an extra bonus mark for this addition feature of yours.

*(BIG Bonus, Optional) Task 4: 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 Neural Network (using Keras, perhaps) 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 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 (exercise code is provided).

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

SUBMISSION:

Submit your PDF file having Taks-1, 2, 3, and 4 in order. Its corresponding source code (if any), figures, and screenshots of running code (if any). Do the above four tasks.

We will discuss more in class

Some Suggestions:

Take a very similar step like Kaggle's Titanic example, 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/, Keras/Tensorflow, alternate Neural Network packages, and many others.). 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 project.

Or you can do all the things in Excel if you want to. It's completely up to you.

Sample blogs:

https://www.kaggle.com/mrisdal/exploring-survival-on-the-titanic/report

https://medium.com/about-developer-blog/xgboost-gone-wild-predicting-returns-with-extreme-gradient-boosting-3e2c16c5bc01

https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb

https://skymind.ai/wiki/neural-network

https://towardsdatascience.com/machine-learning/home