**2023SP - Project 6 - Tower Research - WeeklyNotes05**

**Liao, Yu-Ching**

**Notes:**

* **Log. 2023/02/11 (Sat.):**

After the weekly meeting with sponsors, I soon modify my code so that I can get the score of each exchange. I use the total trading prices that have been clustered as HQ divided by total trading prices to define the score, which can be expressed as “# of HQ prices / # of prices”. The outcome shown as below.

**Table

Description automatically generated**

We can see that 19.csv and 25.csv is acting decent, and 28.csv is the worst. From these scores given, it is highly possible that we can gain the “weights” from them. As a result, in the following week, I am going to do two things: 1. Try to apply “moving error” instead of just “the error on given day”, and 2. Use these scores to create the weights, and then try to apply them on our pricing parts.

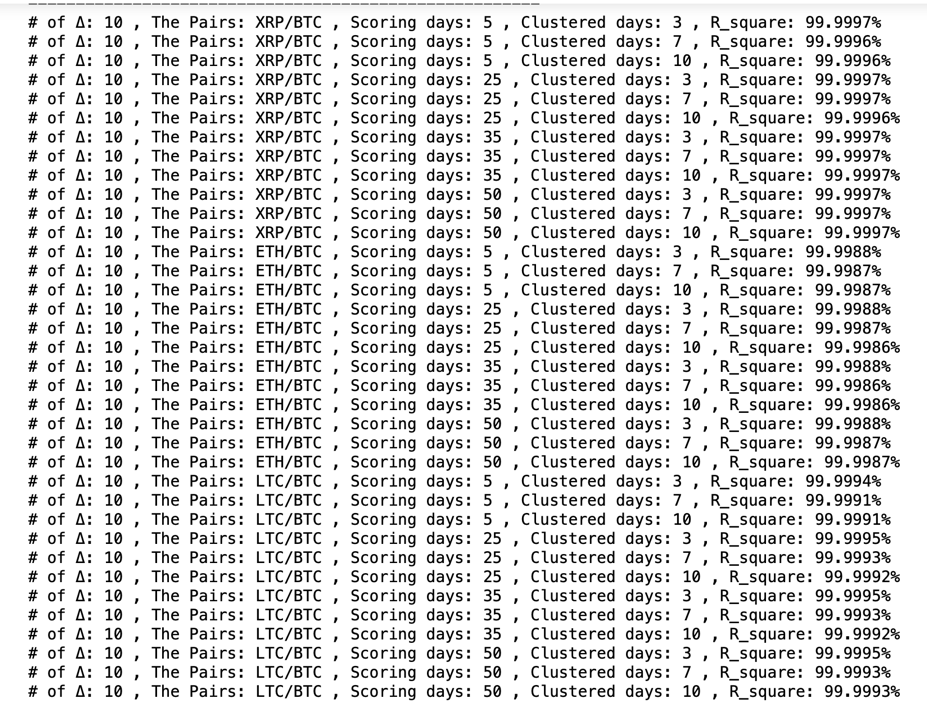
* **Log. 2023/02/13 (Mon.):**

Hyoung Woo’s pricing model seemed to be having a progress. To test his weights is valid or not, I have export the ETH/BTC csv, ETH/BTC csv that only contains High-Quality prices and the ETH/BTC HQ list to him. It is lucky and good choice that I have already constructed the model to store the information, so that I will just have to export the outcome as csv when we need so. After he has tested over his weights and prove that it is valid, we can then combine our models together.

* **Log. 2023/02/15 (Wed.):**

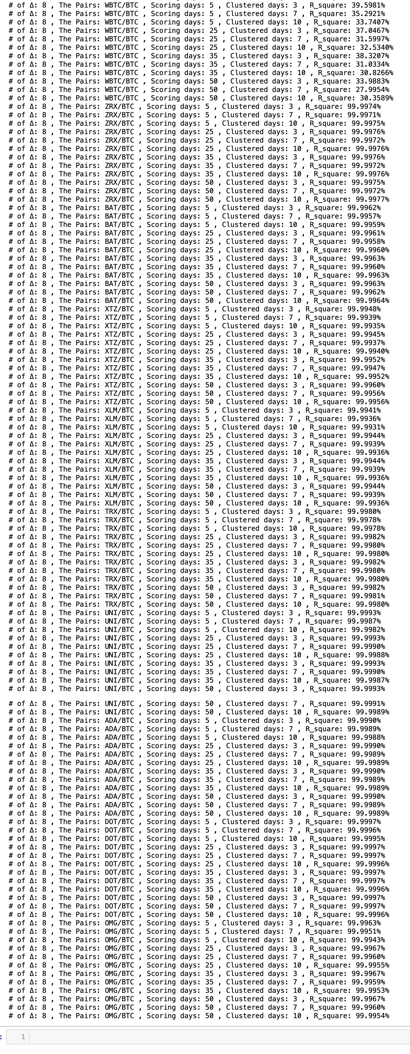
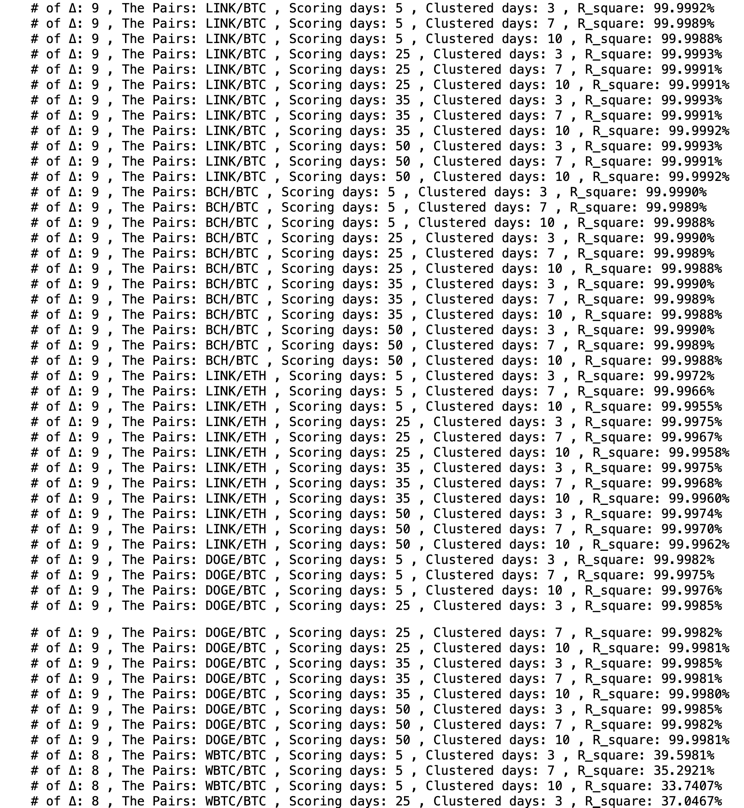
I decided to do something like “tuning” model, that my model will exhaustively test all the combinations of “clustering day” and “scoring day”. “Clustering day” is the days that my model included as the features. For example, if the clustering day is 3, means that my model will cluster on the ***distances of 3 days historical data*** (including today), instead of just on today. And “scoring days” is the days that prices gain their weights. For example, if the scoring days is 30 days, means that the score of given exchanges is gained by how much day that they are qualified as “High”. The way I calculated the scores is using ***the number of days that an exchange being qualified as “High” divided by total number of days it has been clustered***.

I thus tested the pairs that is being traded in 10, 9, 8 exchanges, with the scoring days of 5, 25, 35, 50 days, and clustering days of 3, 7, 10 days.

For pairs that been traded in 10 exchanges, I get result as below:

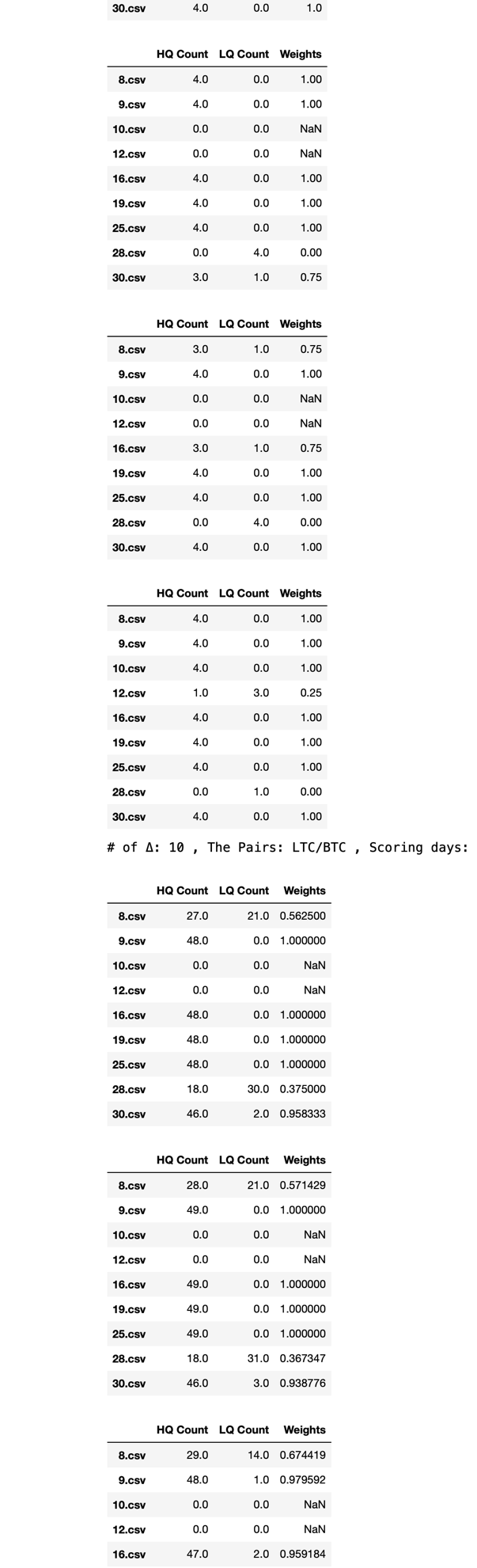
It looks messy since the large amount of data, so I will just make the conclusion. All of them have the R\_square above 99.99%. Though this might seem decent, it is as well possible that it means these results are useless; not able to gain the insights from them. We should check other results.

For pairs that been traded in 9 and 8 exchanges, I get result as below:

****

Most of them have the R\_square still above 99.99%, only WBTC/BTC is not (which is really weird). However, this means that our model performed well at least in those pairs traded in 10 to 8 exchanges. I will then test on 7 exchanges.

Most of the time this week, I was dealing with the bug. Since to deal with cluster model which features are not only one dimension, we will then have to find another way to deal with NaN value. That is, all of the approach I get index and columns will have to be modified. Luckily, I have made the progress on time, and hopefully we can start dealing with millisecond data soon, so that I can test my model.

If we look into the weights (randomly picked but there is not big difference between different parameters), we can see that 16.csv, 25.csv and 19.csv is constantly performed well, which 8.csv, 28.csv and 30.csv is not.

* **Log. 2023/02/17 (Fri.):**

After the meeting, one of the sponsors said he might look into the data for me to see if he can get the insights of solving the problem with WBTC/BTC. And, since all of other clustering model have R-square which above 99.99%, it is difficult to define which parameters are the best. That is, all of them performed too well that we cannot define which the best. But this might not be appliable if we are dealing with millisecond-interval data, since that can be extremely messy. I am so looking forward to see if our model can be useful on the data with that kind of interval.

**Contribution:**

* **Programming:** Cluster with Tuning Parameters.
* **Programming:** Design the model that can test our model on different dimensions.
* **Programming:** Design the model that can test our model on different scoring parameters.
* **Programming:** Exhaustively test all parameters.
* **Programming:** Provide the output that other members need.
* **Providing Idea:** Using R\_squared to see the performance of pricing.
* **Providing Idea:** Using Tuning concept to gain the best parameters.
* **Other:** Presentation of “Cluster with Tuning Parameters” part.
* **Other:** Slides of “Cluster with Tuning Parameters” part.