# MSBD 5013 Report

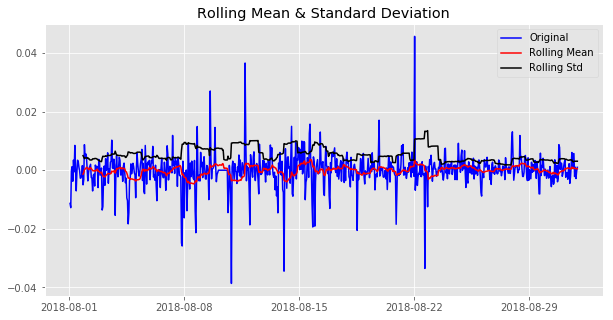
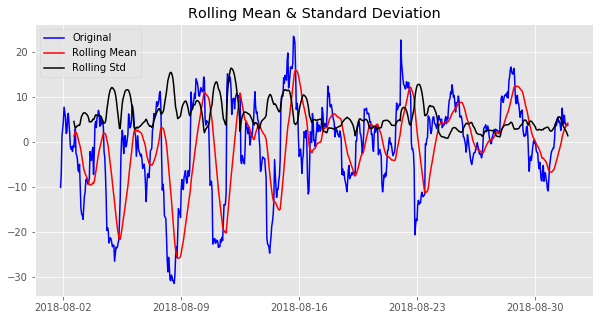
1. **Primary Analysis**

In order to do the prediction, we try to find a seasonal pattern or relation between current and previous price. Therefore, we have run several analyses before building the model. These analyses are mainly focus on the stationarity and seasonality test about the time series from different aspects.

* 1. **Single Crypto Currency Price Analysis**

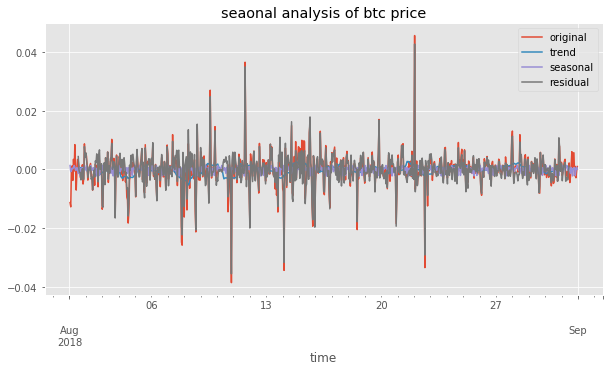
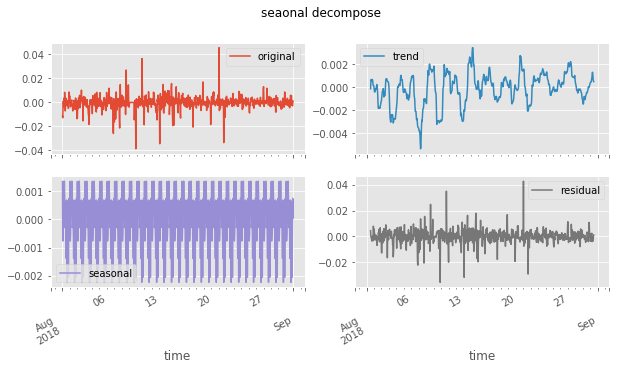
The first and the most direct attempt is simply run the test on the currency data directly. If the data show an obvious seasonal trend during a curtain period, it would very helpful to build the model. As BTC is the most popular crypto currency, we will use it as the target data.

* + - Stationarity



The two graphs above shows the price of BTC during August, the one in right use the log value while the right one doesn’t. The red line shows the rolling mean of the price which fluctuate significantly, even with the log value is unstable. Due to unstable of BTC price, we will use the log value to test its trend and seasonality.

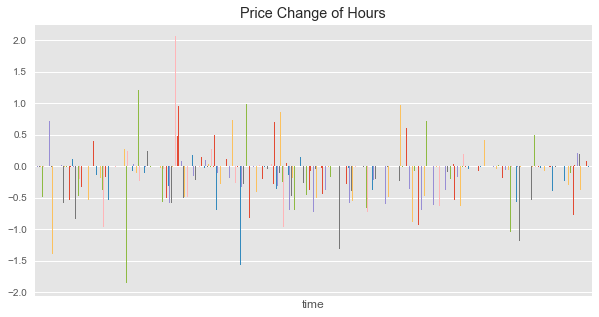
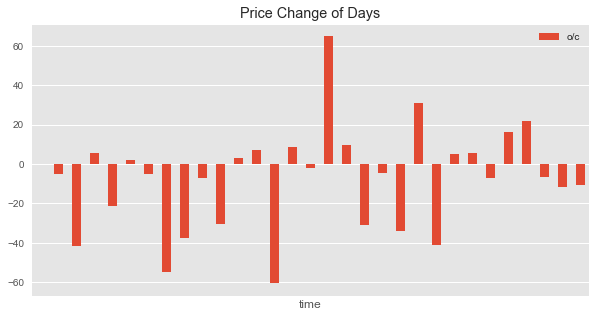
* + - Trend and Seasonality



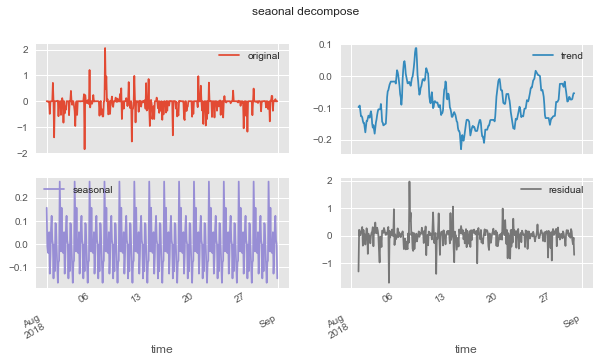
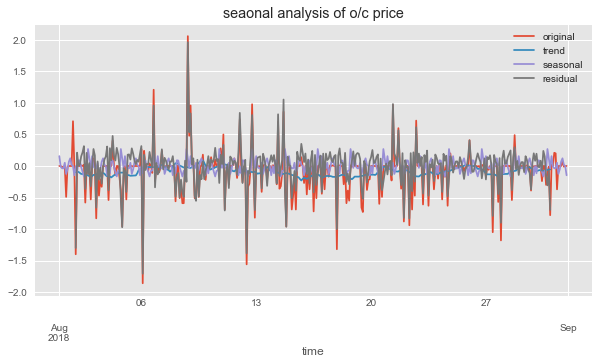
Both trend and seasonal components have small volume, while the residual construct most of the fluctuation. This shows the data don’t show a seasonal pattern, at least in this month.

* 1. **Open / Close Price Analysis**

During the trading, it would be enough for a model that provide a signal of rising or falling rather than a specific price. Thus, the second analyze will try to find a pattern focusing on the price changing in one day. The two graphs below show the changing of max price of a day or hour. It seems have some seasonal pattern.



* + - Trend and Seasonality

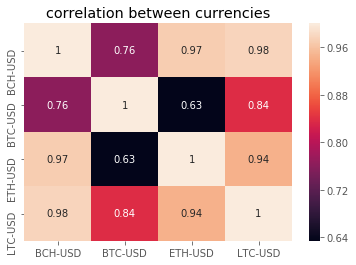
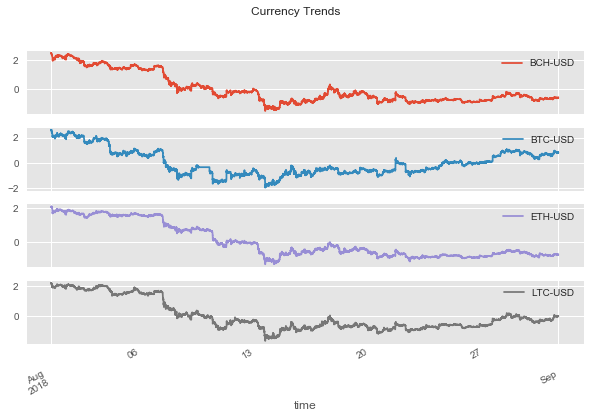


While the analyze result shows the seasonal component don’t make significant influence. Most of the value are contributed by the trend and residuals. These features are hard to provide information to build the model.

* 1. **Crypto Currencies Relation Analysis**

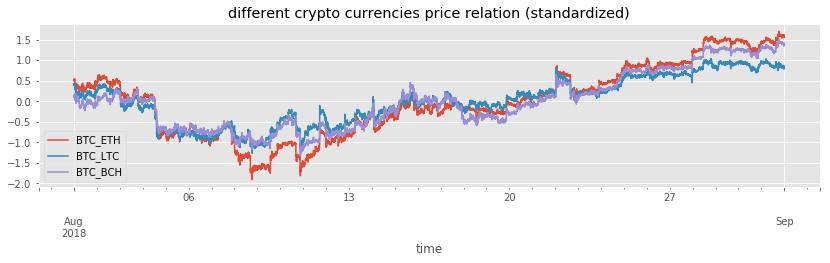
Although it seems no pattern can be found in the currency trend, we still find there is one information might be the breakpoint of the project. We think relation between 4 crypto currencies may have some useful information.

* + - Correlation



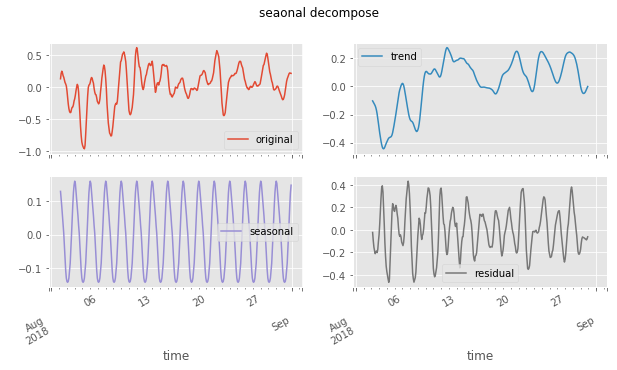
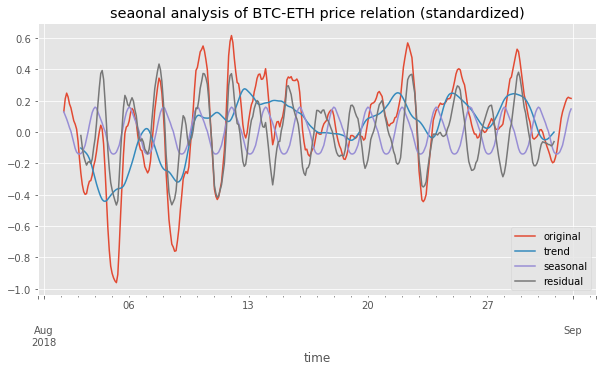
From the graph above, we found there is a strong correlation between the trend of four crypto currency, as well as the price in most of the detail changing. We hope to identify one or two leading currency whose price always change first before others, to function as an indicate signal of the price change. This feature will be very useful for both modeling and strategy.

* + - Standardized Difference Seasonality

First, we standardized the price of each currency. As for BTC occupy most of the crypto currency market, we then choosing BTC as the leading currency to check its relationship with others. The Graph below is the differ during August between BTC and other three crypto currency.

The seasonality analysis is based on BTC-ETH price relation as an example. We can see from the graph below, a relative period stable seasonal pattern, even though the changing volume is unstable.

Despite the residual and trend component take the most part of the total volume, they all show a character of stable rise/fall period. Based on the relatively stable relation, we can build model even only a none-model strategy to do the trading.



* 1. **Conclusion**

Considering the characteristic of the data, the result is just as we expected. On the one hand, it is hard to find any seasonal pattern directly in the price, which is easy to understand. The crypto price is like stock market, there are many influential factors beyond the data itself and most of them are hard to be cachet and integrated in the model. On the other hand, despite the uncertainty in a single currency, the relationship between currencies are relatively stable or at least show a stronger seasonal pattern. The reason is obvious, once there is a price difference, people can use it to make money and the price gap will be filled. As the result, all crypto currency has strong correlation between each other.

Although it is difficult to predict the price directly, we still build several models upon the price and other features generated from it. These models may not have an outstanding performance, but they are different attempts taking different aspects and features. Despite the poor accuracy they may have, the thought behind them might be inspiring.

Based on the thought of ensemble learning, we will finally combine all the weak models, to get a stronger model.

1. **Linear Model**

This model attempt to use the relationships between different currency to predict the price or the trend of a certain currency in a period. In this case, we use BTC-USD as the target predict currency. This part will show several different parameter combinations and comparing the performance as well as do interpretation.

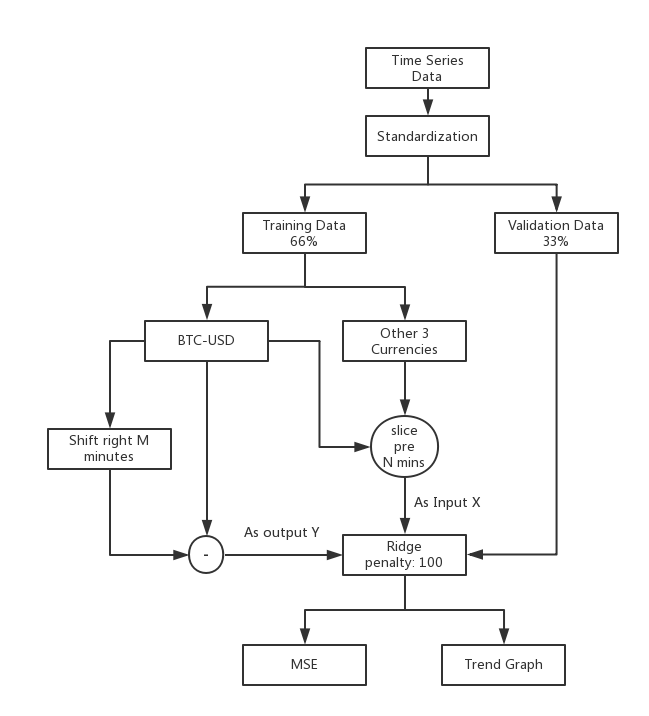
* 1. **Features**

Model Features: The price of 4 crypto currency in N period before.

Model Prediction: The mean price of price of BTC-USD in next 10 minutes.

Here N and M are hyperparameters. For example, we can use previous 5 minutes’ data to predict the average price of next 10 minutes.

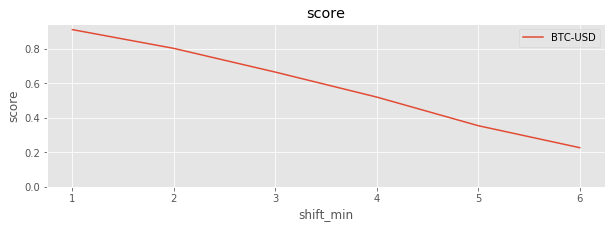
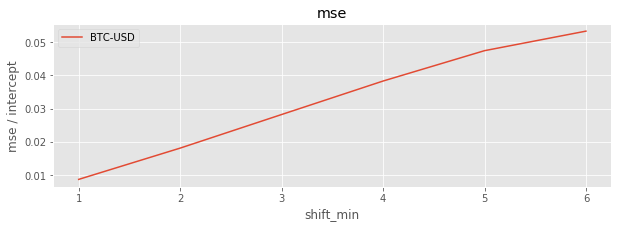
* 1. **Implement**



The modeling process as graph above shows, uses data of all three currencies price from N minutes ago to predict the BTC currency price at one minutes later. Here we have tried two different outputs, one is just directly predicting the price while another is predicting the mean price of next 10 minutes. Since we don’t need to do the feature selection, we choose Ridge as the linear mode with the penalty factor settled as 1 and 100 for 2 kinds of outputs respectively. Moreover, the training data & validation data are separate by 66% and 33%.

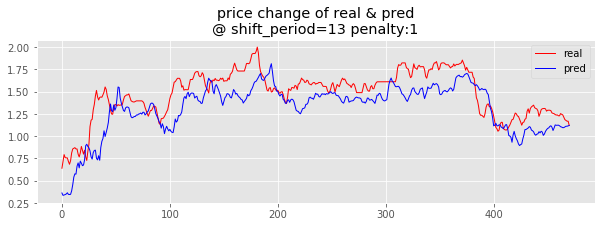
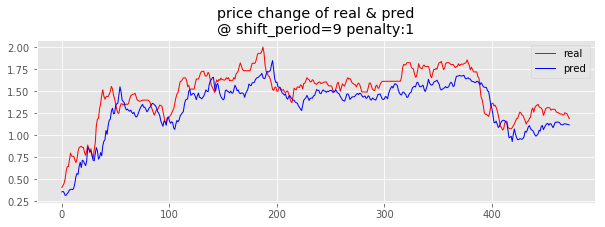
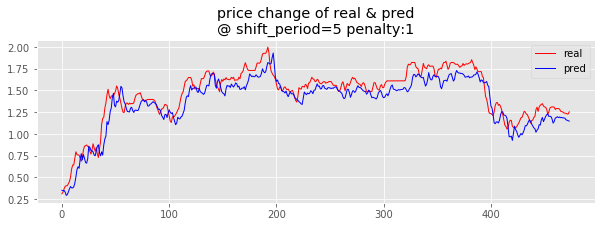
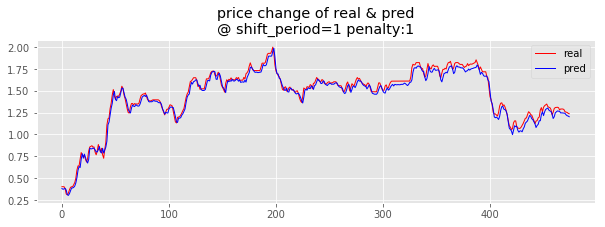
Finally, we use MSE to measure the performance for the regression model.

* 1. **Performance / Evaluation**
     + MSE



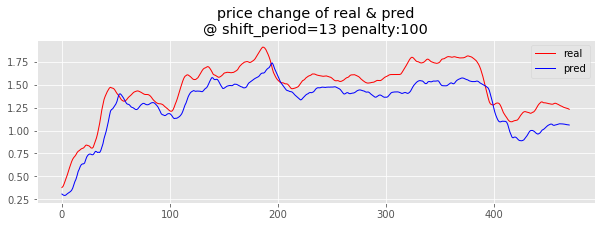
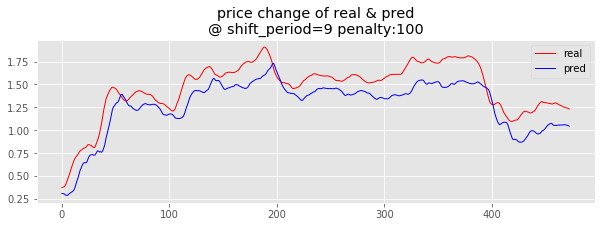
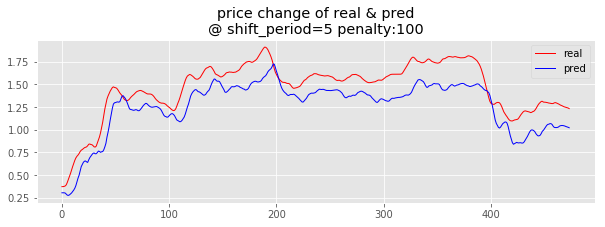
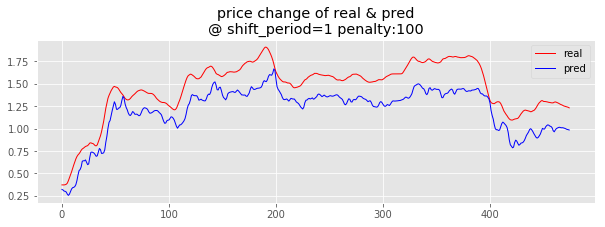
The X axis is the shift minutes of the input data, which increase one unit represents add another four features as input for the model. As we can see, the accuracy will gradually decay by the increase the offset length.

* + - Trend of predict next minutes



From these four pictures we can find 2 things. Firstly, the difference between fitted value and the real trend keeping increase with the larger shift period. While the general trend keeping same all the time. Secondly, just look at the first 2 graphs, the difference between prediction and true value also extend for later time.

* + - Trend of predict mean value of next 10 minutes



While the direct price prediction has very frequent fluctuations which bring many noises to the model. Thus, we use the mean price of the nearest 10 minutes and strong penalty to smooth the curve. As the result shows as above, the more time points we use as input, the smoother the prediction it is. And still the prediction, it keeps the general trend of the real value.

* + - Interpretation

Although the performance seems very good, it can only prove a roughly linear relationship between the four currencies price. In fact, the fitted price curve is guided by the previous true value. As we can find that the blue line which represents the prediction, is always have an offset to the left comparing with the red line. Such offset means the prediction is behind the real value, especially for those turning points. Therefore, it can hardly be used for the strategy to provide long/short single.

However, it doesn’t mean the model is meaningless, we can still generate some features such as the changing rate between currencies, which could have a strong correlation between each other, to do the future modeling.

And in the next step, we have tried to avoid making prediction on the price, we simply swift to a classification approach to predict the rising or falling trend.

1. **Boosting Classification**

As it is not necessary to know the exact price to do the trading, we make a simple classification model to prediction the future trend to support our strategy.

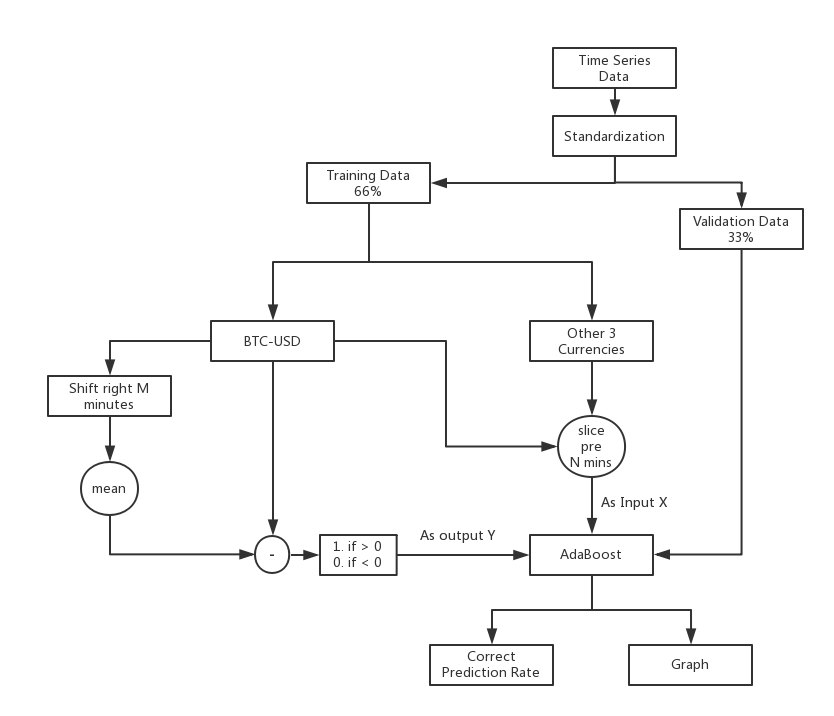
* 1. **Features**

As same as the linear model, we use the same features as input, while simply change the output to a binary value which indicate the rise or down of future price. Since a linear predictor didn’t provide the valuable prediction, we switch the core model to AdaBoost.

Model Features: The price of 4 crypto currency in N period before.

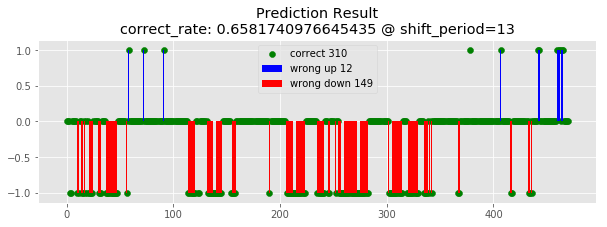
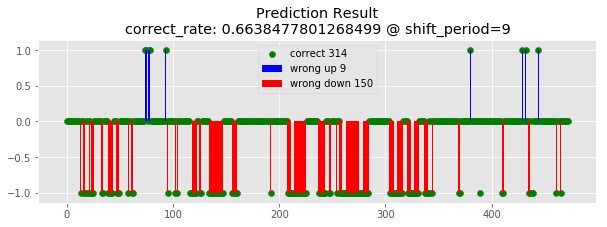
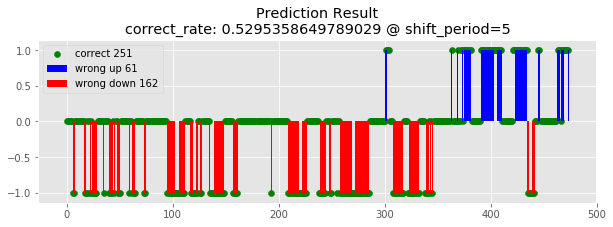
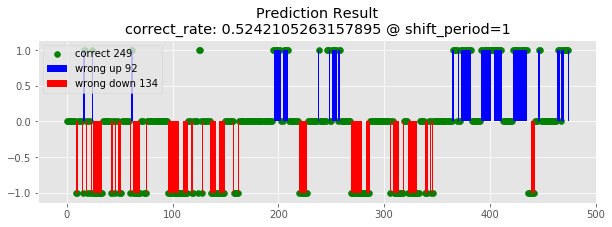
Model Prediction: Raise or fall of the mean price of BTC-USD in next 10 minutes.

* 1. **Implement**



The modeling process is similar with the linear model, with a major modification which transform the output to a binary value. And we use the correct prediction rate to judge the performance of the classifier.

* 1. **Performance / Evaluation**
     + Trend Graph & indicators



The graph shows the prediction results. The red bars indicate the wrong prediction of price fallen, and the blue bar means the error on price raising. Each point represents one prediction and the green dots in the X axis are the correct prediction. It shows with more past data as input, the more accurate the model reach.

* + - Interpretation

After running multiple times of the model, we find the performance is not stable. The result depends on the trending of the currency price in the week. It tends to make more wrong prediction on the opposite direction. Such case can be explained by the model only learn the trend of previous data. If the old data contains more falling, the model will make more prediction as falling, vice versa. Furthermore, the problem of this model is it cannot judge the violent drop or jump, and it treat all price changing as same. But in fact, the wrong prediction will do more damage if the price varies violently.

1. **Random Forest Combined with Grid Transaction**

Random forest is a kind of ensemble method. Its main idea derives from decision trees, which is a popular machine learning algorithm. Below is a sample of decision tree.

图片包含 文字



自动生成的说明

Among all features we have, decision tree method chooses one feature having highest entropy at every split of node. Following this, random forest randomly chooses a subset of features available (usually a square root of total number of features) when splitting a node. Repeat it n times, finally we obtain n different decision trees. At last we average the outputs of these trees as the final prediction result. So next the key part is preprocessing data and set suitable features.

* 1. **Model theory**

In traditional trading theory, grid transaction is a typical kind of trading strategy. Its main principle is called “low long and high short”, which means when we notice price has decreased a lot, we buy; when price has increased a lot, we sell. The criterion is the price edge we set, which seems like a grid. An example is showing below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trend | Price | Positions | [-3%buy, 5%sell] | [-5%buy, 10%sell] |
| Fall | Buy 3 | 90% | 9.1 | 8.5 |
| Buy 2 | 70% | 9.4 | 9.0 |
| Buy 1 | 40% | 9.7 | 9.5 |
| Plain | Base | No change | 10.0 | 10.0 |
| Rise | Sell 1 | 60% | 10.5 | 11.0 |
| Sell 2 | 30% | 11.0 | 12.0 |
| Sell 3 | 10% | 11.5 | 13.0 |

[Source: https://www.joinquant.com/post/539?f=home&m=alg\_example]

From the chart above, we can see that we will change the position when price touches certain edge.

In general, we can conclude that the strategy is according to the extent of price change in the past, we predict if it goes up or down next time stamp. Here the prediction rule is naïve, that is if price has risen a lot, we predict it will decrease. Then we long and vice versa.

Based on this, for a certain currency, we collect the extent of price and transaction volume change of itself and all other currencies in the past as features. Along with these features, we utilize Random Forest to predict if its price will go up or down next stamp. Here we choose one hour as a time stamp. If the prediction result is *up*, we long. If it is *down*, we short.

* 1. **Feature engineering**

Because all data of price is continuous, first we should discretize dataset.

For one-time stamp (60 minutes), if price goes up over 1%, we set its category as *up*. If price goes down over 0.5%, we set it as *down*. Rest are set as *middle*. For transaction volume, if its absolute changing rate is larger than 201%, we set it as *sharp*. Rest are set as *flat*. For one prediction, features are price and volume change described above in past two-time stamp (120 minutes).

For instance, if we want to predict BTC, features needed are listed below:

|  |  |
| --- | --- |
| BCH\_t0: price change of BCH 2 hours ago. | BTC\_t0: price change of BTC 2 hours ago. |
| ETH\_t0: price change of ETH 2 hours ago. | LTC\_t0: price change of LTC 2 hours ago. |
| BCH\_t1: price change of BCH 1 hour ago | BTC\_t1: price change of BTC 1 hour ago. |
| ETH\_t1: price change of ETH 1 hour ago. | LTC\_t1: price change of LTC 1 hour ago. |
| dV\_BCH\_t0: volume change 2 hours ago | dV\_BTC\_t0: volume change 2 hours ago |
| dV\_ETH\_t0: volume change 2 hours ago | dV\_LTC\_t0: volume change 2 hours ago |
| dV\_BCH\_t1: volume change 1 hour ago | dV\_BTC\_t1: volume change 1 hour ago |
| dV\_ETH\_t1: volume change 1 hour ago | dV\_LTC\_t1: volume change 1 hour ago |

Among them, for BCH\_t0, it has categories: up, middle, down. For dV\_BCH\_t0, it has categories: flat, sharp. Rest are so on so forth. Therefore, we can use all features above to train Random forest model.

* 1. **Implementation**

The main problem is setting multiple thresholds including the length of time stamp, bound to discretize price and volume change. As for time stamp, because the target of the model is to predict trend not exactly price, it is not suitable to set a too short length of time. Here we choose 30 and 60 minutes as two candidates. Finally, we choose 60 minutes as it has higher average return across several weeks. Intuitively, it can stabilize exchange procedure in a high fluctuant market.

Another threshold is the bound for discretizing price change. Initially, we hope to lower risk. So we set large threshold. For example, we set -2% as the criterion for ‘down’ and 4% for ‘up’. However, the result is during a week, the program doesn’t even make a position change, which is meaningless in actual trading. After adjusting many times, we choose -0.5% and 1% as corresponding bound.

* 1. **Performance**

We train our model using data upon 2 months (201807, 201808). Here we use cross-validation to check the performance of the model. The accuracy of prediction is listed below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Currency | BCH | BTC | ETH | LTC |
| Accuracy | 0.66 | 0.85 | 0.72 | 0.72 |

From chart above, we can see our prediction can get average accuracy over 73%. In view of high fluctuation of market, such an accuracy can meet requirement.

Besides general test about accuracy, we also combine with complementary strategy to test the model in actual trading. For reducing risk, we add a condition that if cash balance is lower than 20000 USD, we close all positions to make sure we will not be forbidden. We back-test the model in several weeks. Performance are listed below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Time | 0916-0923 | 0923-0930 | 0930-1007 | 1007-1014 |
| Weekly Return | -0.028 | 0.004 | 0.001 | 0.003 |

From the chart we can see that except first week we tested, the whole strategy has a small amount but stable earnings. Because of high fluctuation of virtual currency market, if we can be stable and make money sometimes, we can survive. Preliminarily, the model satisfies the requirement.

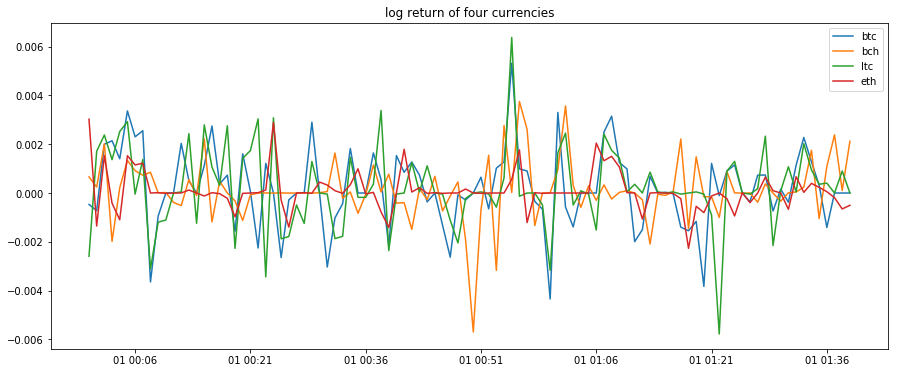
* 1. **Potential optimization**

Based on model above, we can see that several thresholds count a lot. Next, we think that the model can be improved by adjusting these thresholds dynamically according to fluctuation analysis during recent one or two weeks. For instance, if recently the market is relatively plain, we can set the bound of discretizing price change smaller. On the contrary, we can set the bound larger to lower risk.

1. **VAR (vector autoregression)**

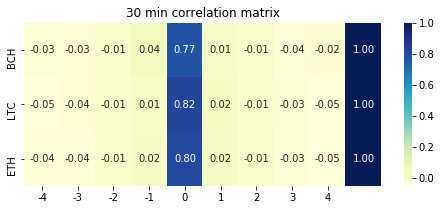
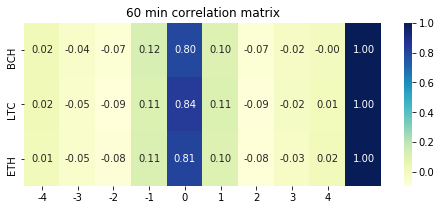
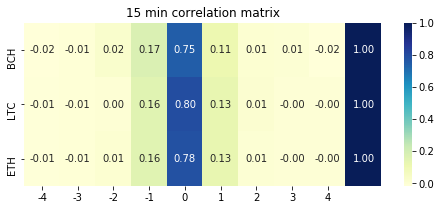
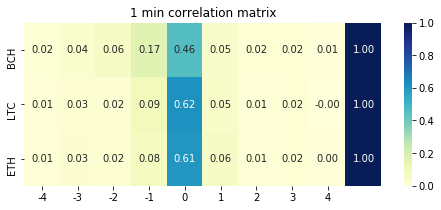
**\*\*Missing Over All\*\***

* 1. **Analysis**

A naïve observation of the log returns of four crypto currencies implies the possible existence of high correlation among different currencies. Below is a plot of the four data’s log return from a random slice of timestamp in 1-minute scale, it is seen that in certain moments, the change in log return of one currency leads that of the others.

To further prove this hypothesis, we investigate the cross-correlation between different currencies, the graph below shows the matrix of the cross correlation between bitcoin and the other three currencies with different leads/lags in 1-minute scales, the x-axis represents the shifts of the other currency in the order of [-4, -3, -2, -1, 0, 1, 2, 3, 4], for example, the element in position [1, 1] of matrix represents the correlation between BTC and BCH shift(-4), and the element in position [1, 2] represents the correlation between BTC and BCH shift (-3), etc. It is seen that the bitcoin has maximum correlation with all other currencies in zero shift, aka in-phase; and that Litecoin has higher in-phase correlation with bitcoin than other currencies.

Expanding the time scale from 1-minute to 15-minute, Higher correlation values are shown in all relationships. This trend continues for expanding the time scale to 60-minute.



A note-worthy fact in 60 minute-scale graphs is that as the phase get closer, the correlation initially drops down and then rise, showing a sigh of seasonality within the patterns.

* 1. **Model building:**

A VAR (vector autoregressive model) is one of the fundamental multivariant models in time series analysis. It could be viewed as the vector version of the univariant autoregressive (AR) model, which has the following general form:



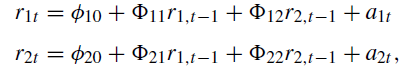
The AR model represents time series values at time t as a function of weighted combination of its previous time values.



In VAR model, all the variables are represented in vector form, a VAR (1) model is in the form:

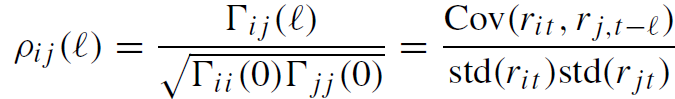
where rt Φ0 rt-1 and at are all of k-dimension, and Φ is a matrix of shape k\*k

when k =2, for example, the expanded VAR (1) model is:



The model shows that the value of time series rk is a combination of the previous time values of all {r1, r2 … rk}

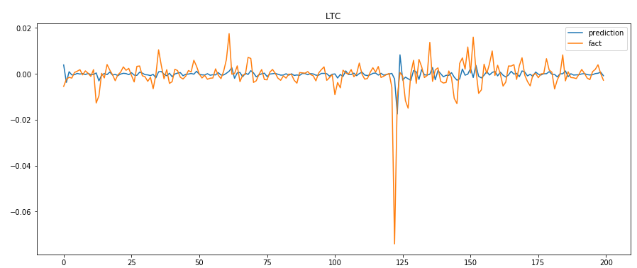
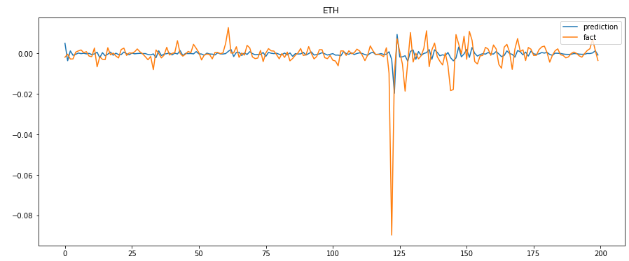
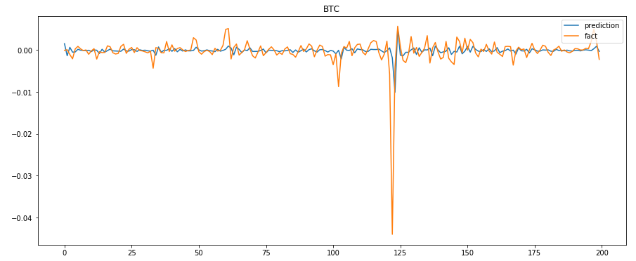
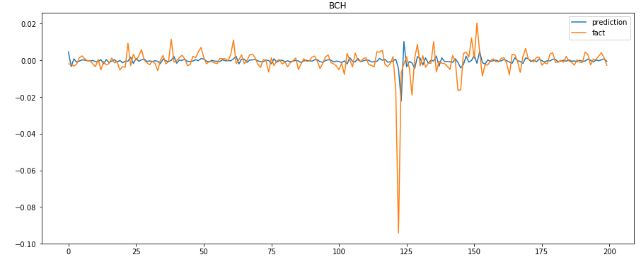
Regarding to our practical implementation, since there are four crypto currencies, we choose k=4 for the model. To determine value of p is determined by the cross-correlation matrix of rr:



In which Γ(l) is the lag-l cross-covariance matrix of rt

* 1. **Performance:**

In practice we fit a VAR (3) model with hourly resampled data. The following is the model’s prediction of log return compared with ground truth in 200-time units



If treating the prediction as a classification task, which means predicting whether the log return is positive or negative at next moment, the accuracy for the 4 kinds of crypto currencies are:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Currency | BTC | BCH | ETH | LTC |
| Accuracy | 0.545 | 0.61 | 0.575 | 0.605 |

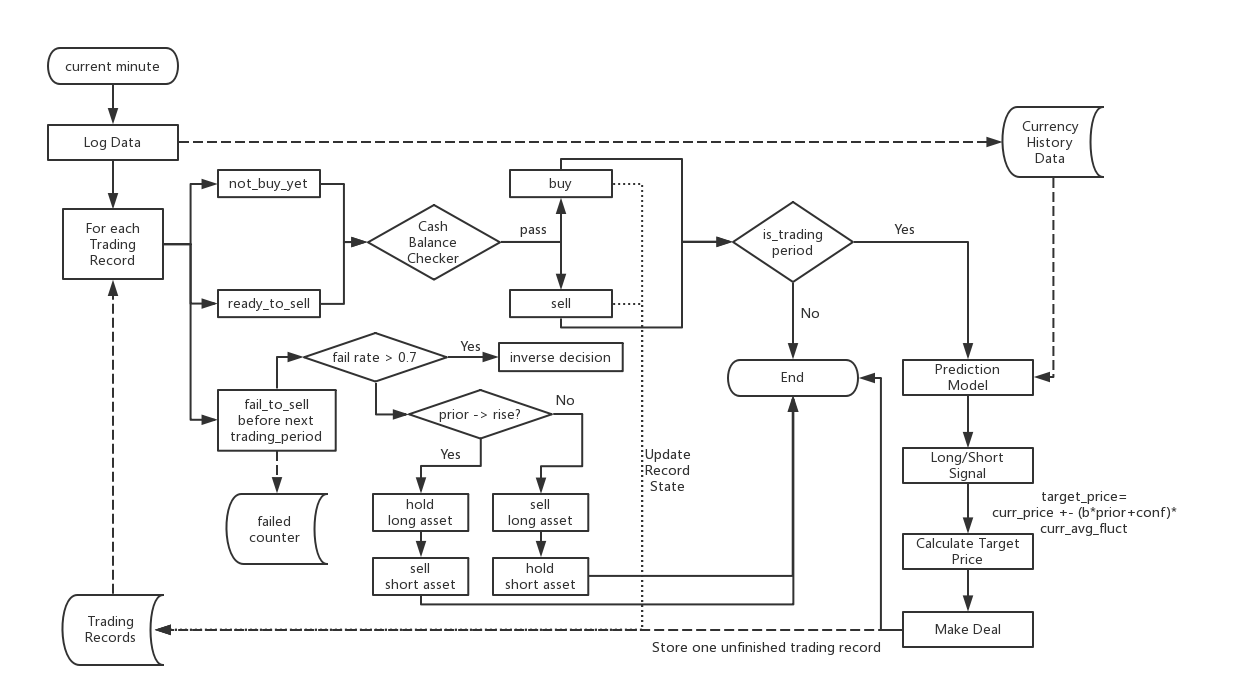
1. **Strategy**

Apart from the prediction model, strategy is another important component. It should not only decide the timing and quantity of long and short, but also need to monitor the balance level in case it is too low to continue the trading. Furthermore, due to the low accuracy of the model, the strategy will need to have a mechanism to do the correction or even handle the corrupted model as well. This chapter will introduce our strategy from these aspects.

* 1. **Details**

Comparing to the simplest strategy, which is completely passive to the long/short signal, one of the advantages of the active strategy is it have a mechanism of stop-losses and a proper money management which will try to make the best use of all the resources.

The process graph above illustrates our strategy. In each minute, we will first store the currency data to support the model making prediction. Once the system judges current time as the trading time, a prediction period unit (30 mins~4 hours) of previous observed data will be sent to the model and get a prediction. Normally the prediction result will be transferred to a probability of currency trend as the long/short signal. Based on the probability and other pre-settled factors, we will calculate a target price and generate a trading record. The formula will consider both confidence, average price fluctuation of the specific crypto currency and a prior judgement of the trend of next week made by ourselves. Such prior will also play a role to handle the dirty trading (those cannot be sold before the next prediction period).

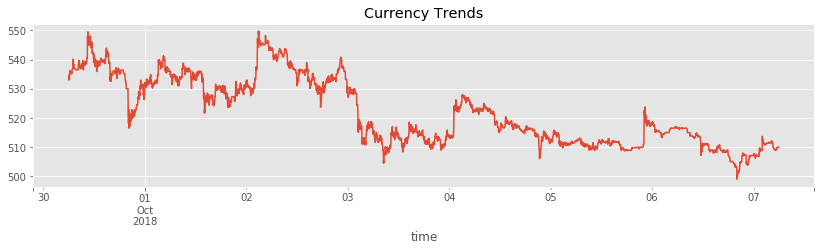


Whether this trading will be success or not depends on the next minute price. On the one hand, in order to avoid the influence of noise (meaningless fluctuation), we will only make new prediction and trade after a prediction-period which is defined as the time span the model required. On the other hand, during the trading gap, we will keep monitor the price changing, once the price reaches the target price of a trading record, the strategy will automatically sell and close the trading. Here by sell means short the longed assets or long the shorted assets, and it depends on the trading record type. While every change of the asset balance needs to pass through a cash balance checker to make sure the cash balance will not lower than the limitation to avoid the framework stop any further trading.

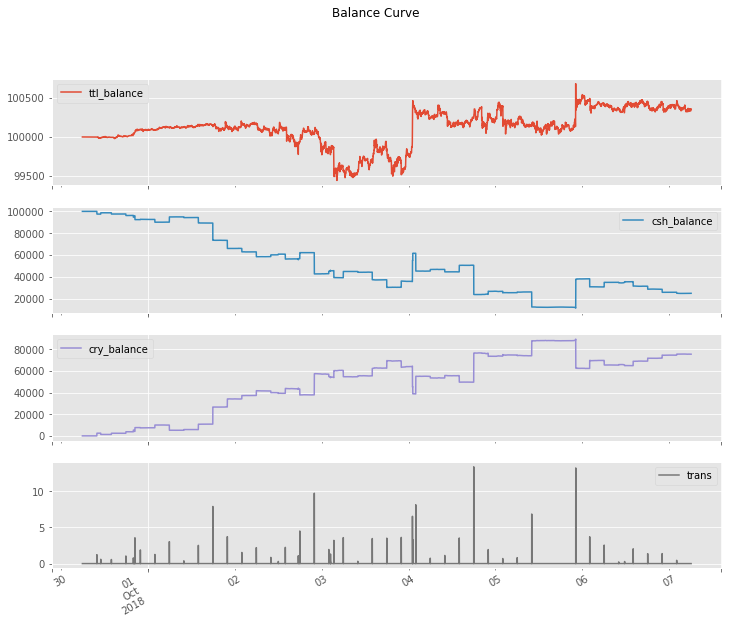
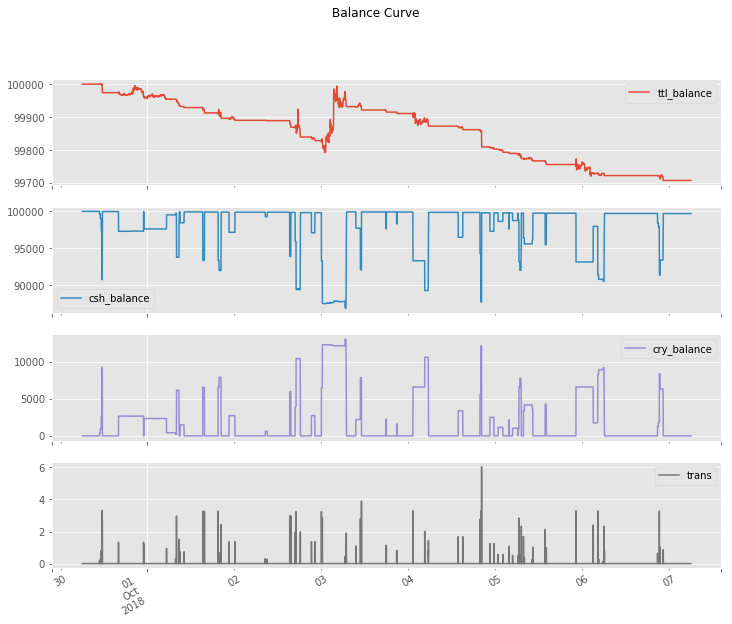
However, not every trading will be normally closed. It is inevitable to have some failed trading left after one period due to the wrong prediction or the unsuitable target price. In such case, the system will count the failed times, after a certain number of trading, if the failure rate is higher than 70%, it will trigger the prediction flip mechanism. Furthermore, how to handle the failed trading will depend on the pre-settled prior. If we think there are more chance that the currency will rise in next week, all longed assets will be hold till the end while the shorted assets will be sold immediately to stop losses, vice versa.

The existence of these mechanism will help us avoiding many unnecessary losses, making up some short comes of the model, and to improving the trading performance.

* 1. **Evaluation**

To illustrate the performance of the strategy, we choose the data from Sep 30th to Oct 7th. And making a comparation with one of the demo strategies.

The graph is the currency trend in that week, as we can see have several big wavs and show a general trend of decrease in final.



The graphs in the right is the balance curve of our strategy, while the left is the three-white-solider demo. Comparing them two, it is obvious that the former one reduces many unnecessary trading which bring a relatively continues increase of the total balance during the stable stage of the currency price. Although there is still violent changing during the whole week, these shortcomings are mostly due to the inaccurate prediction which is less relatively to the strategy.

Furthermore, this strategy also tries to make the best usage of all resources while still keeping the cash balance in the safe level. On the contrast, the demo strategy only uses 1/10 cash during the whole process. To achieve this, we have invested in all four crypto currencies at each trading period and using the specific trading unit respectively.

Comparing to the demo, the strategy finally results in a positive gain during the decrease of the market.