COMMON MISTAKES IN STOCK PRICE FORECASTING

Experiment Findings · June 2019

1 author:

Tim Dong
Rochester Institute of Technology

COMMON MISTAKES IN STOCK PRICE FORECASTING

Foreword

For the past two months, I have been trying everything I could think of to improve my stock direction prediction model. Despite many research papers that report 60% to even above 90% accuracy, I found this task to be extremely challenging to give practical, usable results. I found that many research papers have suffered from at least 1 or more of the mistakes that I will explore below. My earlier submission also suffered from the last mistake, and thus I would like to change my project into “Common mistakes in stock price forecasting”.

Abstract

In this paper I consider the task of binary classification, but the idea should apply to regression task as well, not only for stock price, but also for any time series. Forecasting the stock market has long been considered one of the most challenging task. Nonetheless, there are many research papers, even from prestigious universities, that claim to achieve the prediction accuracy significantly above random, ranging from 60% to above 90%. To the best of my knowledge, no model has successfully passed all of the test to be a useful in practice. All proposed models suffer from at least one type the mistakes that this paper attempts to reveal. As my goal is not to undermine any researcher but to help people aware of common mistakes that may render their models completely useless in practice, I did not include any research papers as evidence. Although I do not include all of the papers that I reviewed, readers can easily find many papers that attempt to predict the stock markets and verify the results for themselves.

Notation

There seem to exist a confusion between the word “predicting” and “forecasting”. While “predicting” may refer to in-sample as well as out-sample predicting, “forecasting” usually refer to forecast the unseen (out-sample) values. In this paper, both “predicting” and “forecasting” refer to the out-sample only, as a model is only useful if it can tell something a human does not already know (the future).

To keep it consistent with my earlier work, I will denote *step* as the trading/forecasting window. That is, how long into the future that we want to predict. For example, we want to predict the price difference between today and the next 20 days, then *step = 20.*

N: Total number of samples

Ntrain: Number of training samples

Nvalidate: Number of validating samples

Ntest: Number of testing samples

(*N = Ntrain + Nvalidate + Ntest)*

Common mistakes in Time Series forecasting

In this section, I list 5 most common mistakes found in many research papers that try to predict stock price direction. The list is ordered in terms of the ease to spot the mistake.

1. **Using "hit ratio" (or "accuracy) as the performance metric**

Hit ratio or accuracy, which is the percentage of correct predictions over the total predictions to make, has long been considered inappropriate for classification tasks. Thus, many other performance metrics, such as confusion matrix, F1 score, Matthews Correlation Coefficient… were developed as its alternatives. Nonetheless, this mistake can still be observed in some papers. This mistake is the most obvious and has been explored in many articles.

2. **Shuffling the data**

Shuffling the data is also an easy, yet common, mistake. A time series must obey its temporal order, shuffling it destroys this characteristic and causes the mistake of using future values to "predict" past values, resulting in arbitrarily high performance for the model.

3. **Normalizing the test set**

Normalizing the data is very common in any machine learning task. Yet, one must be cautious not to normalizing the whole test set as doing so will "leak" the information about the future distribution into one's models. At any given point in time, we can only normalize the data that we use to train our models, normalizing the unseen data causes look-ahead bias will certainly result in worthless good results.

4. **Not using sliding/expanding window**

Normally in a machine learning task, one could split the data into train/test or train/validate/test set. I claim that this is a very common mistake in time series, especially stock price, forecasting. It is a fact that stock price is not stationary, even with differencing, it is not guaranteed that stock price can be made station. The stock statistics will undoubtedly change overtime, thus by training the model on the train set and making predictions on the test set, we are asking our model to write in English while teaching it Japanese. Thus, the "lazy" method of simply splitting the data and do all the training, validating, and testing on separating sets is not only impractical but also unnecessarily undermine the model's performance. This mistake is very common as splitting data is a standard technique in machine learning and it does not seem to introduce any look-ahead bias.

5. **Using ALL the training set**

This mistake is the main contribution of this paper (my earlier submission also suffers from this mistake). Although it seems obvious that one should use all the data in the training set, I claim that doing so also introduce look-ahead bias. The reason is at any given moment, if we have *Ntrain* samples/observations in the train set and we want to predict *step* periods into future, we only know the result (label/price/value) of *Ntrain – step* samples. We must exclude the *step* number of samples. Concretely, consider the case with *N = 120, Ntrain = 100, Ntest = 20, step = 4.* That is, we have 120 samples, we use 100 samples to train, 20 samples to test, and for any given sample, we want to predict the closing price 4 days into the future. In other words, we have two arrays: *train[1:100]* and *test[1:20]*. Can we use the whole *train[1:100]* set to predict the next 4 days closing price of sample *test[1]*? As suggested above, the answer is **No**. If we do so, we "hinted" the model about the price that should not have been available. To predict the target value for sample *test[1]*, we only know the target value for *train[1:100-4] = train[1:96]*. Using the sets *train[97:100]* would require us to know the price of *test[2:5]*! To see this clearer, consider how we will use our model to trade in real life. Let *t* denote the current day, using the convention training scheme, to predict whether we should buy or not, that is, to predict the price difference of day *t* with day *t+20*, the model needs the label for day *t-1, t-2,...t-m,* with *m* being the number of look-back days. But then how can we know the label for day *t-1*? To have the label for day *t-1*, we need to know the price difference between day *t-1* and day *t+19,* which is clearly some day in future.

As an empirical results, when I use all of the training set prior the a test sample, I get very good result (80% to above 90%) as demonstrated in the presentation. As soon as I use only the data with labels that is available prior to a test sample, the result fluctuated wildly and eventually approached 50%, which I discussed earlier when I mentioned the effect of excluding the *step* days samples from training.

Conclusion

It is surprising how many senior researchers failed to take into account the above mistakes. In writing this paper, I call for more attention from the academia into the practicality of their research. A research can very well be theoretically sound but not applicable yet, but suffering from logic that render it completely useless is another story; a story that must be addressed. It is not enough to write a research with "significant" results to earn a degree or getting it published, a research should be both theoretically sound and has real-life application.

I claim that the presented mistakes can be found in almost any papers that attempt to forecast a time series, skeptical readers are welcome to verify this. I would appreciate any feedback regarding these mistakes.