Linear Regression Trading Algorithm

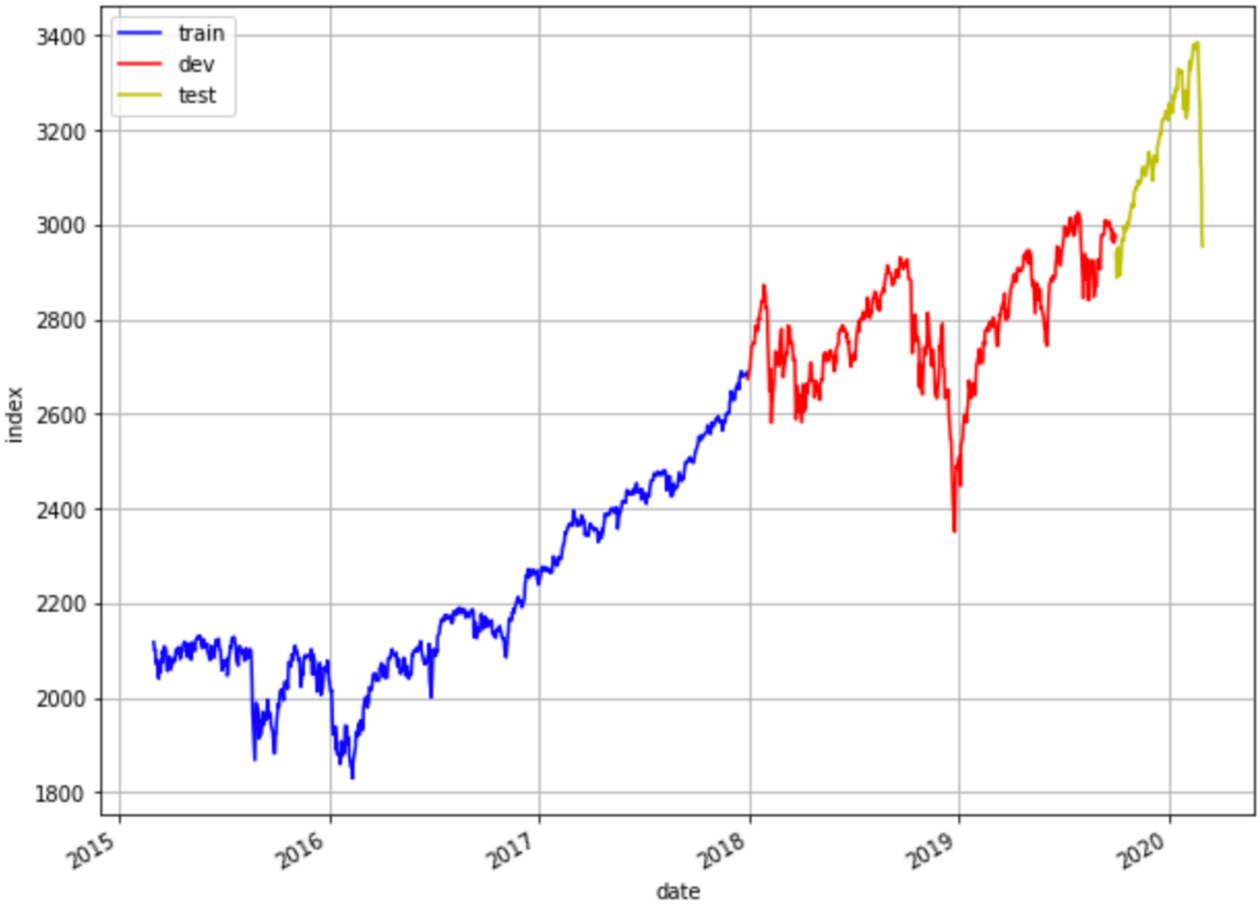
Prediction Method

The algorithm is using linear regression to predict the index value in next day. We are using values of the last N days to train a model and predict for the N+1 day. Testing different N from 1 to 30, we can find the optimum N for the best prediction.

Prediction Result

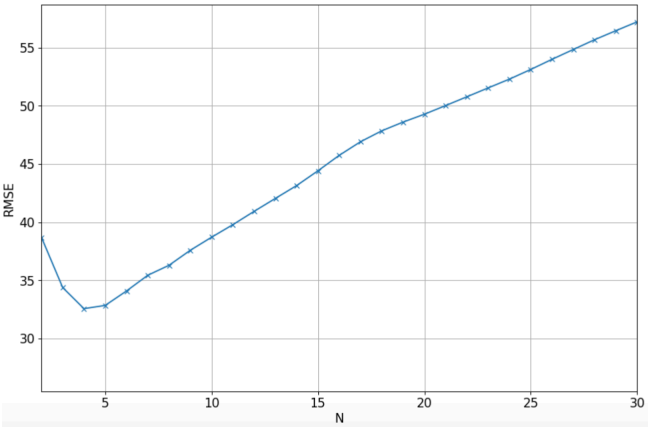
• Data

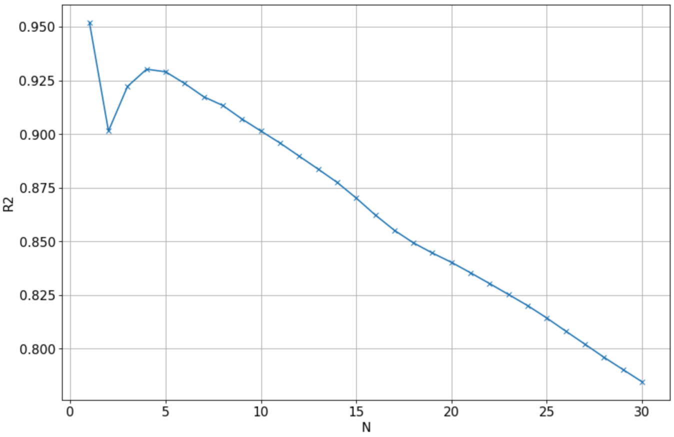
We chose S&P500 index data to test our model, with the time ranges from 2015-02-28 to 2020-02-28. The test set contained the data after 2019-10-01, and 35% of the whole data were marked as dev set. The rest were train set.
num_train = 715
num_cv = 440
num_test = 104

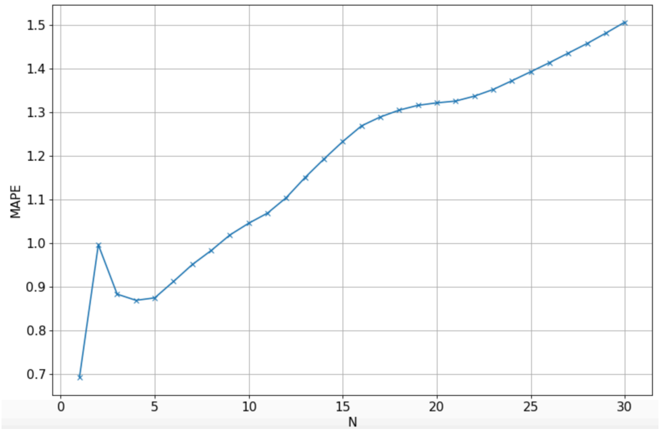


- **Training Result**

Using our model, we trained the data and get the values of RMSE, R2 and MAPE with different N.

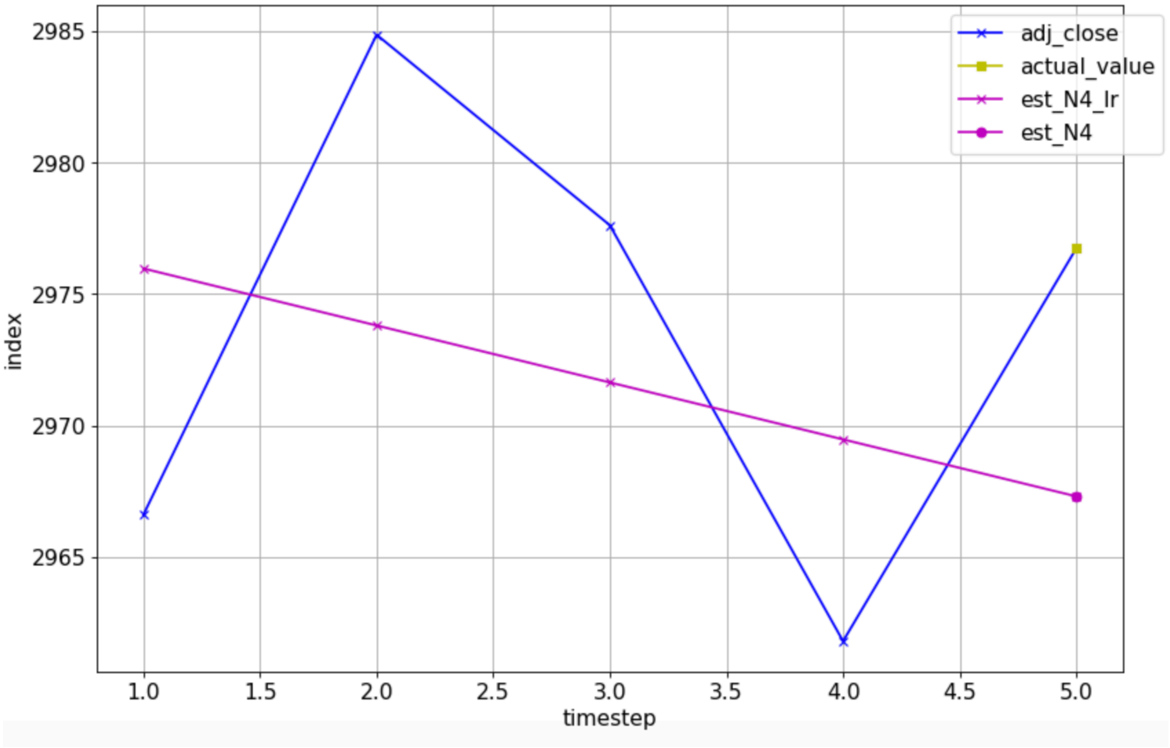




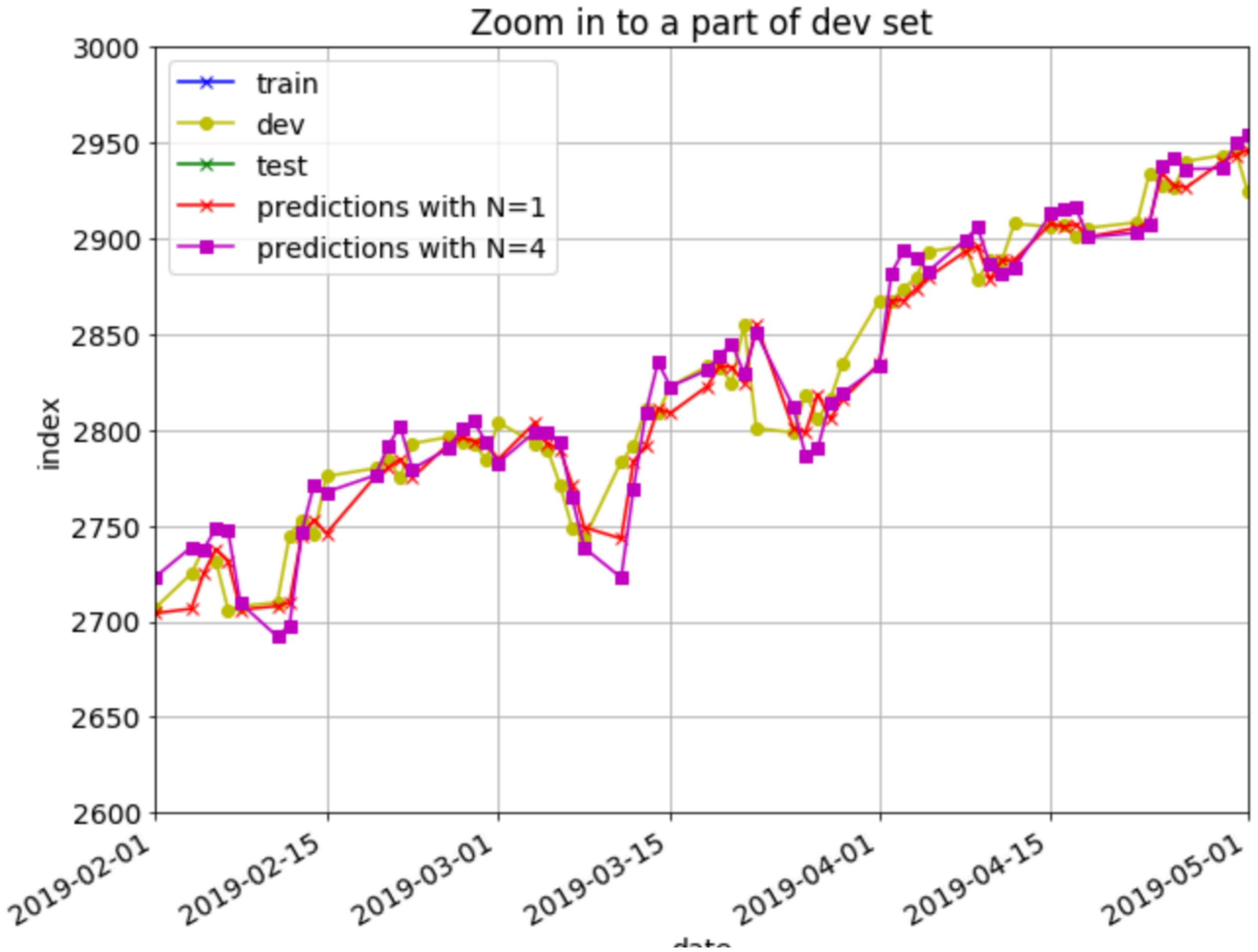


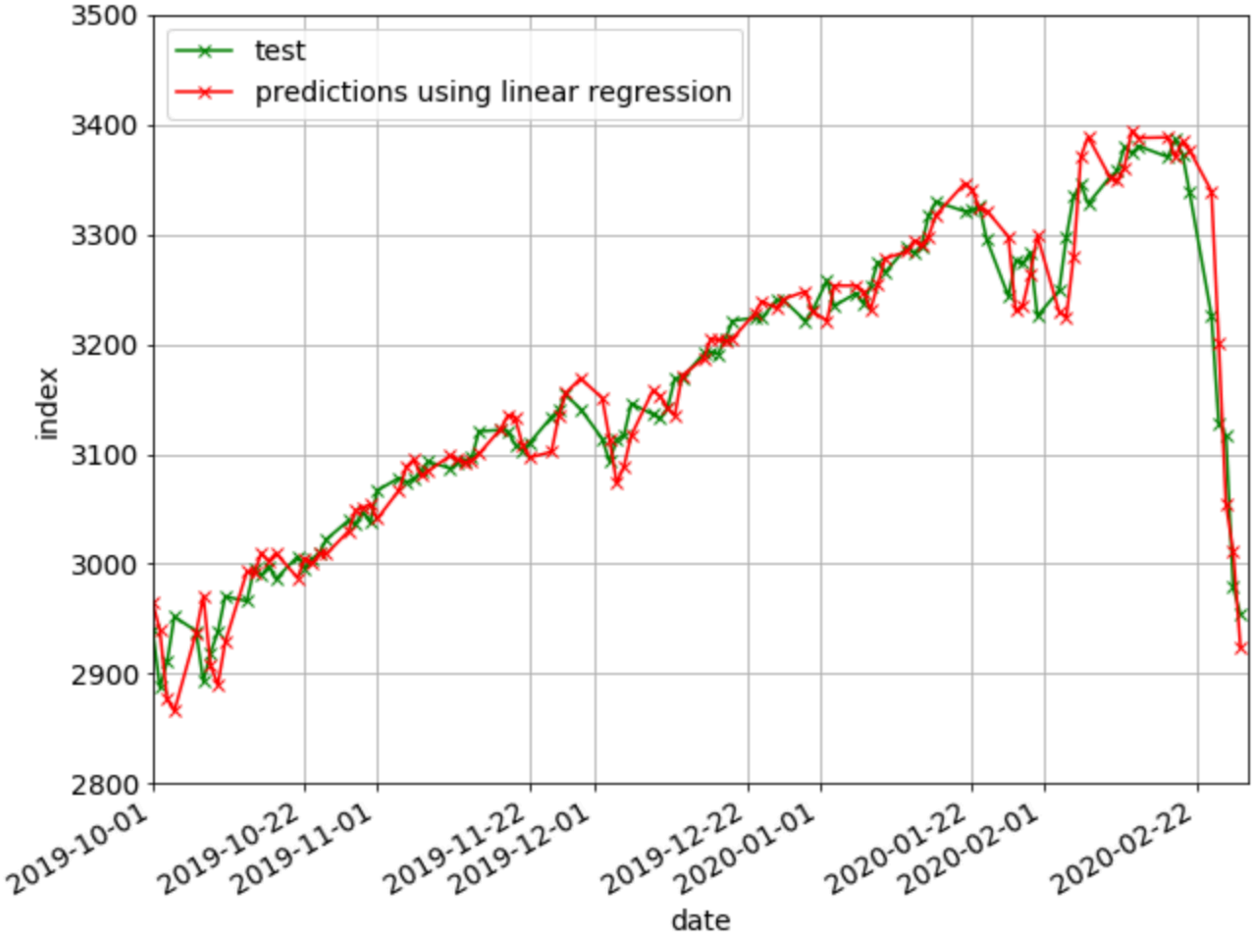
Roughly the optimum N is 4. The lowest RMSE, MAPE and the highest R2 are achieved with N=1, followed by N=4.

For the rest of this algorithm, we use N=4 to predict.
We can check the prediction with a specific day (2019-10-31):



To check its result with more data, we can see the prediction with a part of dev set and the test set.

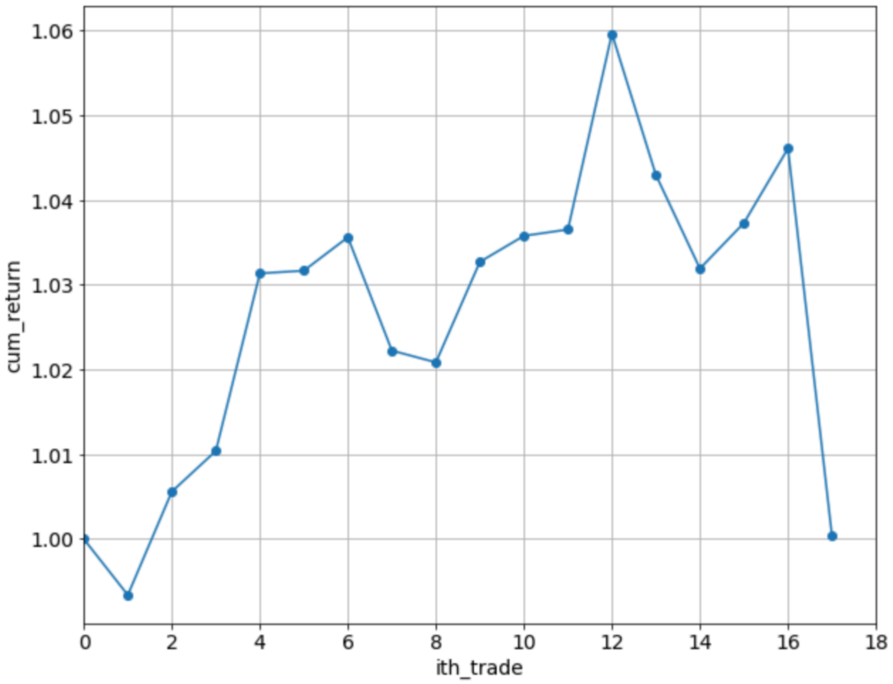




It shows that the algorithm can predict the price trend roughly, but it maybe not that precise for a single day. We can also see that in the test set. Even for some dramatic change of the index, like the drop in late February, 2020, the model successfully predicts the tendency. The model has a RMSE of 30.394, a R2 of 0.948 and a MAPE of 0.701%, which is a relatively good result. However, it might not predict well for a particular date, and we will test this in our trading algorithm.

Trading Algorithm

We use test set data to trade. The principle is that we buy when the model predicts a rise in the index and sell when it predicts a fall. Suppose there is no short for our trading.
In our 104 days, we buy and sell for 17 times, and the profits were recorded. We calculated the cumulative return and the win rate for this method.



The algorithm has a win rate of 64.7%. It shows that the trading principle's win rate is acceptable, but the cum_return doesn't have a good performance. We think the problem is because of the win-loss ratio and that we have no stop-loss. From the figure, we see that the average gain for each trade is much less than average loss. The last trade, for example, causes great loss and is not stopped.
Thus, it is possible to use this model to predict the price and construct a trading algorithm based on it. But to make it effective, we need some more auxiliary method to control the loss if the model "make mistakes".