Stock trading with machine learning

ISM-E1005 Forecasting Methods in Business Analytics

Santeri Hynninen

Motivation and research questions

Trading algorithms have always been of interest to mine. I had heard many stories about Jim Simons and Medallion Fund that has outperformed markets consistently. I'm not planning to pursue a trading career, but I want to understand and learn more about industry and methodologies. This paper aims to find could Finnish small cap companies offer any trading opportunities with relatively simple approach with machine learning.

Problem setting

As discussed in lectures, usually the simpler you can formulate the problem the better. Thus, we will not try to predict exact stock prices, because that would be a way more challenging task, and as an investor the exact price might not be as valuable as the direction of change. Of course, larger wins would be nicer, but as risk averse human beings not losing money with some gains would probably be preferred option. Our forecasting model tries to determine in which direction the price will develop in the next month. We would also like to evaluate the probability of the prediction, if possible, because we are looking for buy or sell signals, we would only want to invest into the most prominent predictions that we expect to bring us gains with the lowest risk possible.

Data gathering and exploration

We chose 5 companies for the training and testing. The companies were chosen from the Nasdaq First North exchange because we had a hypothesis that companies with smaller market value would not be as closely analyzed, thus offering better opportunities. The past performance of the stock was not investigated, nor do I own any of the stocks. Most of them were names that I had heard before and even considered investigating. The 5 Finnish stocks were the following:

Company	Ticker	Days of data
Spinnova	SPINN.HE	919
Admicom	ADMCM.HE	1762
Betolar	BETOLAR.HE	801
Lapwall	LAPWALL.HE	718
Titanium	TITAN.HE	1847

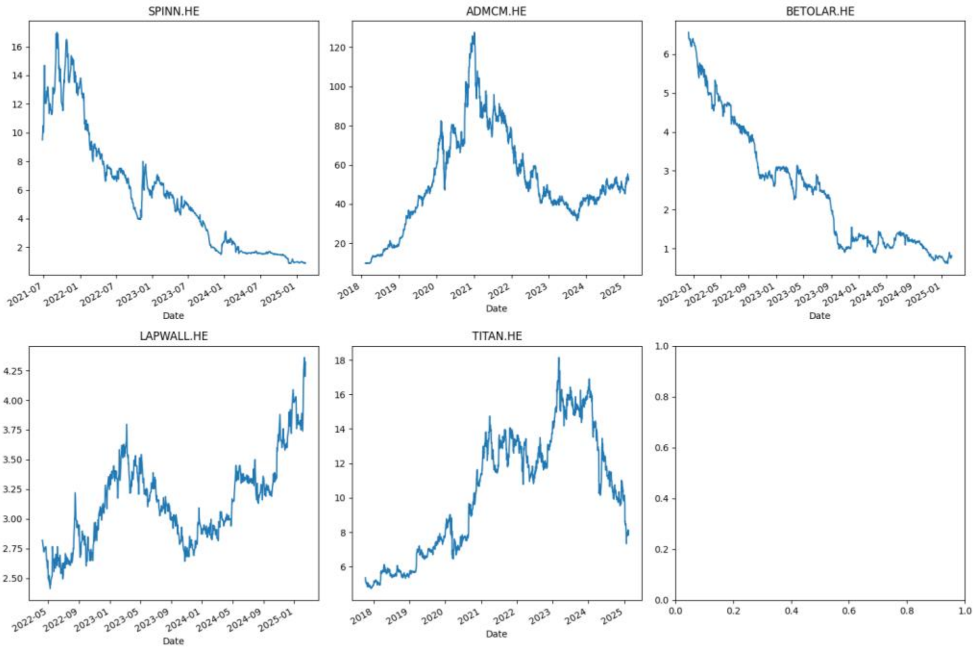
The data was gathered by using the yfinance library that scrapes the public APIs of Yahoo Finance. They do not claim that the data is 100% accurate nor should it be used for online trading, but for our purposes, it was great enough. We gathered daily values of open price, highest price, lowest price, closing price, adjusted close and volume. We chose to include all available data, but some of the companies had rather recently had their IPO, thus the amount of data was limited. The amount of data can also be seen from the table above.

We also tried to better understand the stock price development for each stock, and descriptive statistics were used to evaluate these. First the daily prices were differentiated by removing the previous value and chosen statistics calculated. The mean return was very close to zero for all companies, also the standard deviation was rather small for all other companies but Admicom. Spinnova and Betolar especially had significant skewness in the returns. The Kurtosis values showed leptokurtic characteristics on all other stocks than Lappwall. These metrics well identify how large stock movements usually are if they are biased towards positive or negative side. These could be also compared to the model's performance and if these could correlate with the results.

Ticker	Mean	Standard Deviation	Skew	Kurtosis
SPINN.HE	-0.009370	0.268661	1.427813	14.617712
ADMCM.HE	0.024134	1.419608	0.306314	8.626359
BETOLAR.HE	-0.007200	0.077421	1.122467	9.351904
LAPWALL.HE	0.002092	0.061276	0.443866	1.142543

Ticker	Mean	Standard Deviation	Skew	Kurtosis
TITAN.HE	0.001464	0.194466	-0.238716	4.692107

Also, the graphs were investigated, and possible cycles, trends or structural changes were tried to be found. The stock prices can be seen below.



As we can see many of the stocks have clear trends and structural changes. Spinnova had strong but short uptrend after the IPO but that changes to a strong downward trend. Admicom had also a very strong bull run, which transformed around as a strong bear run. Recently it has shown small positive signs. Betolar has had very strong downtrend for their whole time after IPO. Lappwall had strong uptrend after the IPO, which cooled down but as again picked wind. Titanium has similar situation with Admicom, but they have not shown any positive signs yet. There have been major changes in the macro-economic environment that have affected many smaller companies which might explain the strong bear markets and structural changes in the stock prices.

Model selection

In order to better understand how we should treat our data before modelling, we have to choose the model itself. So, the data will best serve the needs of our model. Because our target was binary, many statistical methods would not serve us well, also we expected non-linearity in the data that would exclude many linear models. We chose to approach the problem by using machine learning that was relatively easy to implement, could handle binary classification well, and also offered many possible models to choose from. Furthermore, we chose to use decision tree-based models due to strong performance with non-linear data, and ability to handle large number of features, without extensive regularization nor would the tree models need complex feature transformation. Also, because of our previous understanding we chose XGBoost that is a boosted decision tree model, which most often achieves way better results than singular decision tree.

The rather simpler one model approach could be criticized but due to the scope of the project and the problem itself not other models were implemented. Also, the need for probability excluded many models for this project. Some other models could have been considered, for example SVM and neural networks, but on the other hand, SVM would need scaling to be implemented, and Neural networks are a lot more complex to implement. Also, our objective is to create a concept model for learning purposes not the absolutely best model. We can also point out that XGBoost gives good overview of the importance of different features, which can help in improving the prediction model.

Preprocessing and feature engineering

A number of steps were conducted in preprocessing and feature engineering to create dataset that would bring value to us. First of all, because all stocks had strong trends, we did not want to use the stock prices as they would change trough out the time, and the rules that the model would have made would not be valid. We adopted using differencing to combat this problem, which was done by subtracting the previous value from the newest value. We also conducted numerous other features based on historical data.

Category	**Feature**
Basic features	Differentiated Open price
	Differentiated Close price
	Differentiated Volume
Recent volume and price changes	Daily direction streak
	Lagging values 1 to 3
Ratios for the day	Low vs high price
Trend and moving averages	Price ratio of max price in past year
	Price ratio of min price in past year
	Volume ratio of min volume in past year
	Volume ratio of max volume in past year
	Moving average of daily streak from past month
	Moving average of differentiated close price
	Moving average of differentiated volume

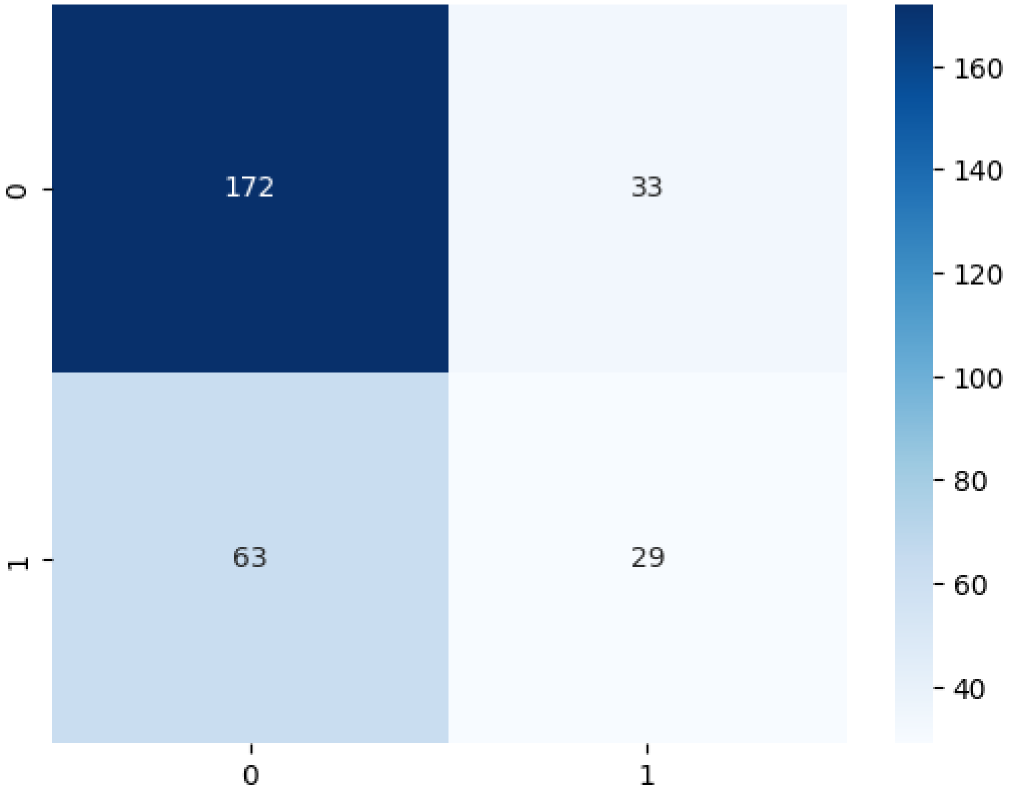
Modeling and Evaluation

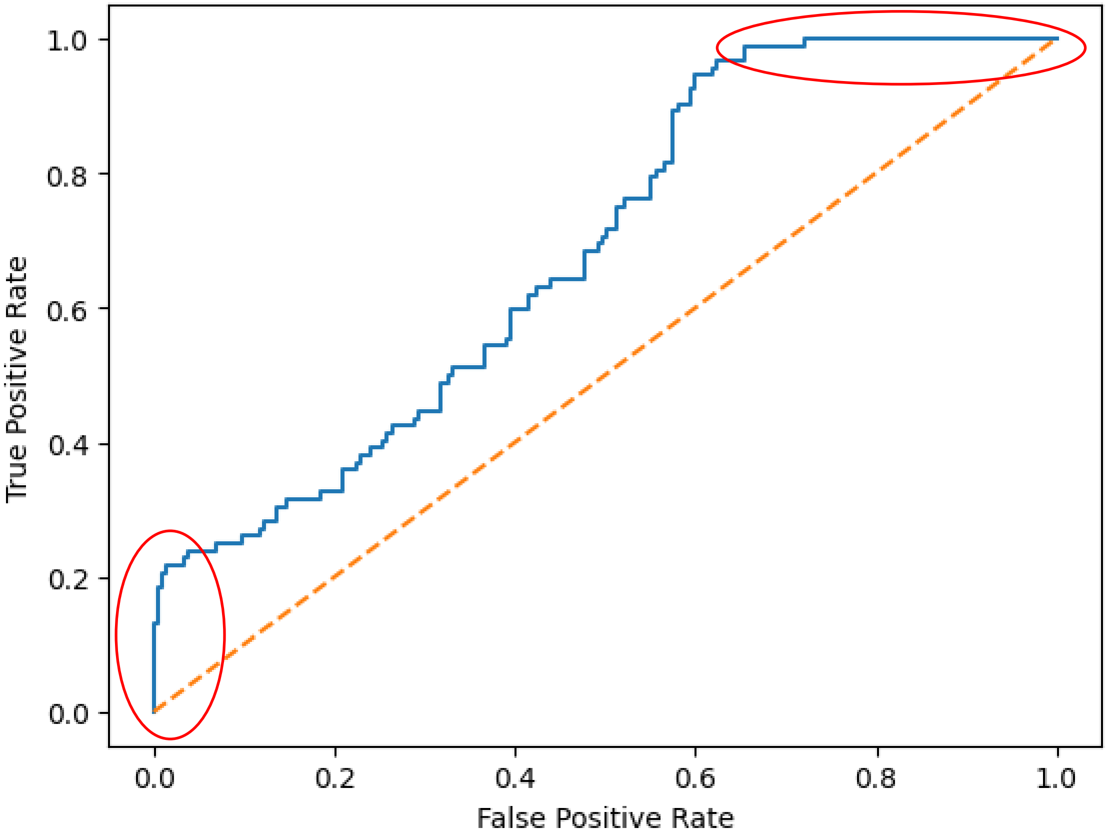
Time-series split with 1/3 test size was done to train the model with 2/3 of the data and evaluate the model with the last third so we could evaluate how well it performs with unseen data. XGBoost was initiated with 50 iterators and set random state. No further hyper-parameter tuning was done on this point, because the benefits for performance are usually small and XGBoost usually performs relatively well with default values.

With the Spinnova the model performed relatively well, and thus it is used as an example. First of all, the classification metrics were looked at, which did not indicate strong performance:

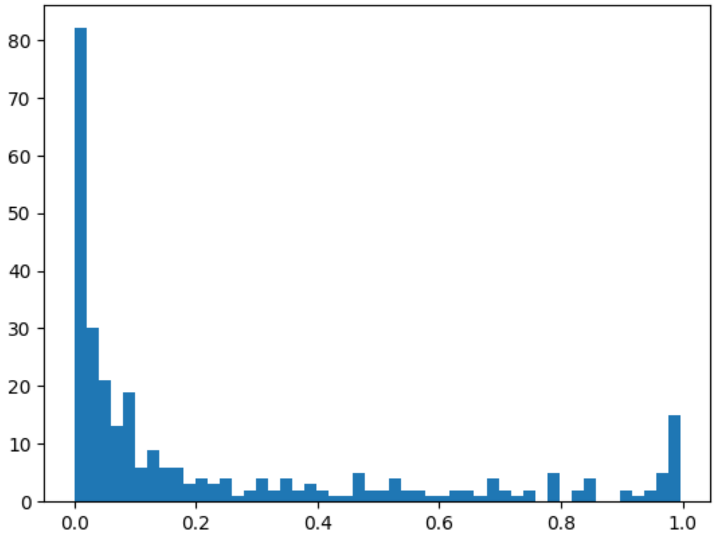
	Precision	Recall	F1-Score	Support
0	0.73	0.84	0.78	205
1	0.47	0.32	0.38	92
Accuracy			0.68	297
Macro Average	0.60	0.58	0.58	297
Weighted Average	0.65	0.68	0.66	297

The accuracy of 68% was appropriate in this kind of task, where anything over 50% can yield positive results. But on the other hand, when comparing this to the price graph, the model could just use the trend as an advantage and be biased towards negative returns which the option 0 would indicate. Also, the difference in metrics between the two options would indicate that. When comparing the results to the confusion matrix, it was clear that the model was biased towards negative returns. It had chosen option 1 less times than guessed option 0 wrongly.

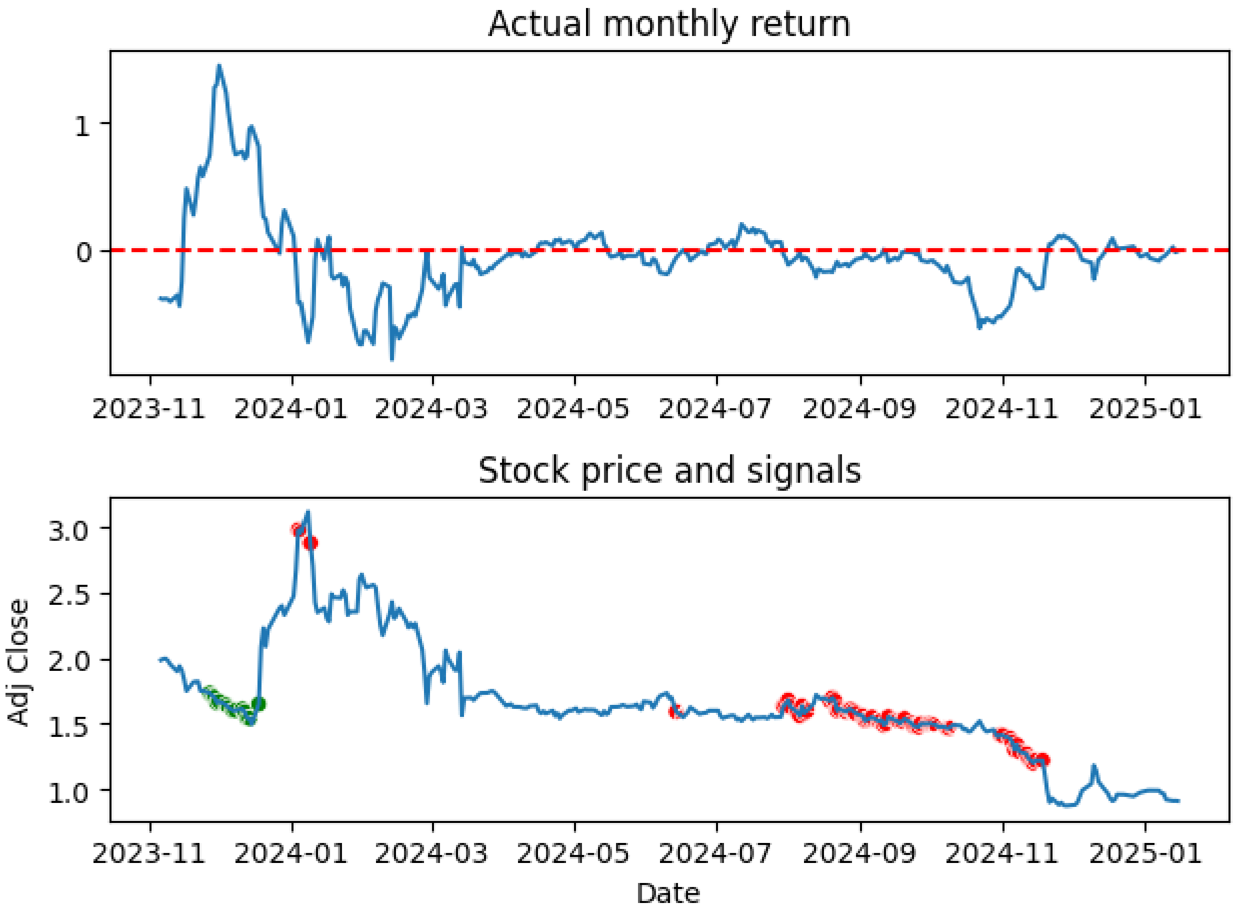




In further investigation we graphed the ROC curve, that showed that the model was performing better than random guessing. In our case the ROC curve gave us very useful hints about the model performance. The threshold could be set so that the model would be 100% right with the test set if we would set the thresholds accordingly.

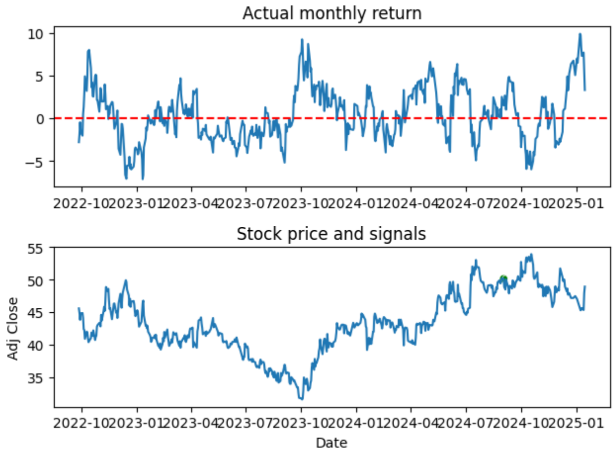


When evaluating the histogram of prediction probabilities there were clearly many cases where the model was not sure about the answer, but in some cases, it would be 100% sure about the returns of next month. Thus, we set the thresholds so that it would predict only cases where it would be 100% sure about the prediction. These results were graphed together with price development:

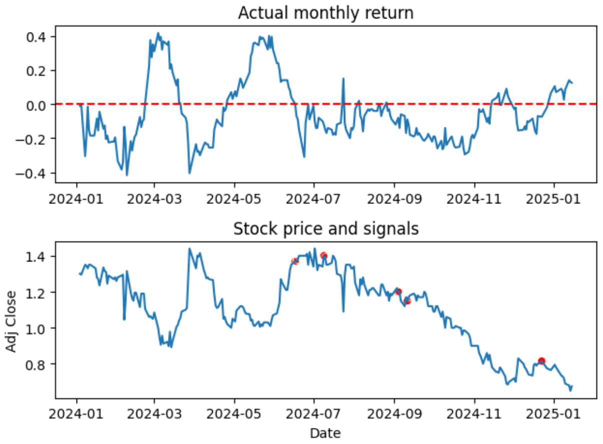


Now the performance of the model seems a lot better than previously thought. It shows continuous buy signal before a large price jump, and even signals sell at the very top of the model. It also predicts continuous sell signal before a larger drop in the price. Of course, the model does not predict every opportunity with the price, but the ones that it showcases are promising and would have produced positive gains without even taking advantage of the evident trend of the price development.

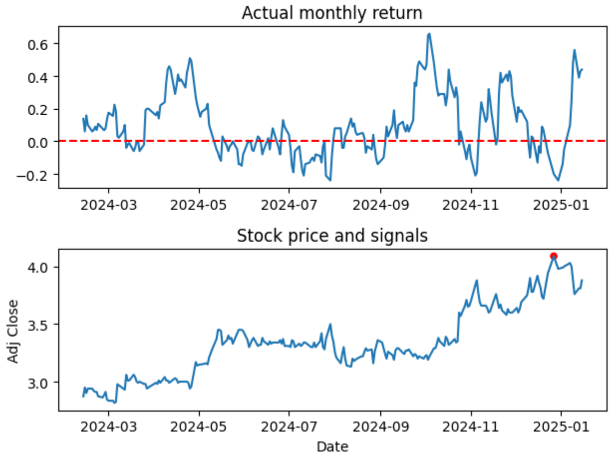
Further on, all other tickers were run through the program and the following signals were predicted:



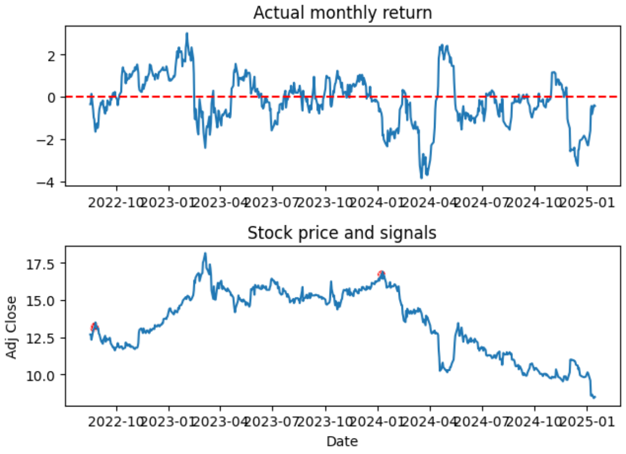
Admicom



Betolar

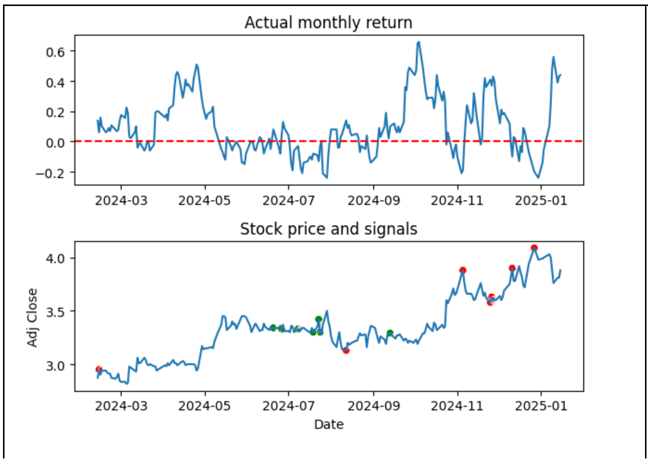


LapWall

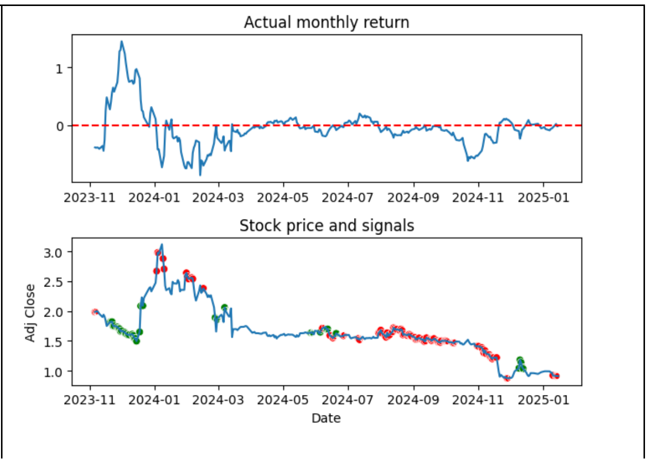


Titanium

Based on the other stocks we can see that our model more often gives negative signals than positive, and the signals are very sparse. This can be adjusted based on the number of iterators of the model and what our error margin should be. For example, adjusting the error margin to 5% on both sides. Predict way more signals to the LapWall but with more error. Same happens with Spinnova with strange buy signals that are not well timed:



LapWall(With 5% error rate)



Spinnova(With 5% Error Rate)

Discussion

Our model is rather simple concept model, and would need way more tuning and testing to be viable option for trading. On the other hand, it showed some solid signals, that could have produced gains if opted to invest. Also, we have to note that our model sees only the most used stock metrics that do not measure how well the company performs, what is the macro-economic situation or investor sentiment towards the stock. Thus, the model misses a lot of information that could and will affect the stock price. On the other hand, it can spot spots where there is upside or downside pressure.

Many further development ideas could be adopted, because the data is limited cross-learning could be adopted to spot common patterns from stocks that have the same kind of market conditions and characteristics. Also, more models should be tested, and they should be hyper tuned to give the best possible performance. When evaluating the model, we should probably split the data into 3 parts: training, validation and testing. The threshold customization should be conducted in the validation part, so we wouldn't just match the threshold based on the performance in testing data. Lastly the data and features could be engineered and developed, rather easy ways could be incorporated into the index or peer data to give the model benchmark about the common market movements.

This project very well taught the essential components of time-series forecasting like time-series splitting, need for differentiating and critical evaluation of the performance. The approach was less traditional but helped the writer to learn more about trading algorithms, and how they could work with rather simple data and model.