Directional Stock predictor

Summary

The notebook in this repository contains a variety of different machine learning models which use technical indicators from the past to determine whether a given stock will go up or down over a given time in the future. The idea for this is inspired by [1]. The indicators used in the models are the following:

1. Measure of stock momentum
2. Stock RSI
3. Stock volatility
4. Measure of index momentum
5. Index RSI
6. Index volatility
7. Measure of average volume of stock
8. Measure of average volume of index

These indicators are generally calculated for the past n days, upon which the model makes a prediction about the stock price after m days in the future. The index indicators are calculated using an ETF of the index for data purposes.

Using $m = 10$ and $n = 20$, we get an average model accuracy of around 60% for the deep learning model and SVM although the deep learning model is slightly higher. We don't expect that much discrepancy as we have a limited number of features for deep learning to really see an advantage. This is an area I hope to improve.

We can implement these signals into a simple trading strategy. If the model predicts the stock will go up and we are not already long the stock we buy it. If however, the prediction signal is that the stock will go down and we are long then we sell our position.

Experimenting with a variety of different m values, I found that the return didn't vary significantly, backtesting on a variety of stocks, if we changed m. For a constant model accuracy we expect smaller m to offer higher returns as we pick up on finer upward rallies of the stock. However, as m increases we see a generally higher accuracy of the model and so the effects of less possible return

and increased prediction accuracy cancel out roughly but nonetheless this is an area to explore further. Below is a plot of the returns of the strategy for $m = 10$, $n = 20$ for a given stock in the S&P 500 and a plot of the optimum strategy, which shows the returns if the model was 100% accurate.

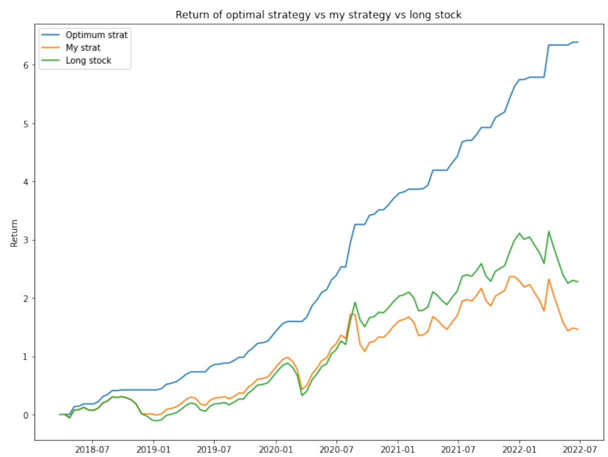

Figure 1: Backtest of return on AAPL stock of systematic trading strategy using $n = 20, m = 10$.

We see that we actually make less return than if we were to just hold the stock and this comes down to the fact that most stocks in S&P 500 have a tendency to increase as time goes on. As such the models learning from this tend to often predict an increase in the stock price and as such the strategy has very similar returns to the stock. This is until there is some news in the market which causes a sharp jump in the stock that the model may not have predicted. This means that it misses out on the extra return and in fact may have to pay more to buy the stock at later times, causing the return to drop even further. This is what is happening in the figure.

Overall, the model currently does not seem to present a profitable trading strategy, we need to significantly increase the accuracy of the models to be able to guarantee some form of long term returns. However, sharp unpredicted jumps in stock are make or break for this strategy in the long term and failure to predict these correctly can lead to less returns than just holding the stock.

References

1. Madge, Saahil, and Swati Bhatt. "Predicting stock price direction using support vector machines." Independent work report spring 45 (2015).