Volatility Voyage: Assignment 3

Introduction

In the last meet, we discussed the importance of risk management strategies and how they can save you from facing losses in trades which the strategy which we coded fails to identify. In fact, this is a very common scenario in real life too.

Imagine you're planning to go out for the day and you check the weather forecast. The forecast says there's a good chance of rain, that's like your strategy giving a signal to take a trade. It's based on data only, looks promising, but it's not 100% certain. Say, you decide to go out anyway, just like you'd take the trade.

There can be two ways of action now :

1) You trust the forecast and don't carry an umbrella. If it rains, you get soaked.
2) You believe that there's still a chance it'll rain and carry an umbrella just in case.

The umbrella here is your risk management, like a stop-loss or limiting how much money you risk. Though we invest much time and money on optimising the strategies to give better predictions/signals, it is risk management which adds an extra protection to the you/capital.

Implementation

Strategy :

You now know how to code a trading strategy and backtest it. It's time to explore deeper and build something more complete and polished. Your task is to design a trading strategy that combines the strengths of different styles. You can take elements from momentum-based strategies, mean reversion techniques, volume signals, volatility-based filters, or any other approach you've come across. You are free to use various strategies. Combine them thoughtfully into a single strategy that you think is robust and realistic.

Risk Measures and Backtesting:

Once you've created your strategy, the next step is to integrate risk management techniques into your backtesting engine. You can use the references below to guide you regarding a few techniques, but it is expected from you to explore new ideas of risk measures and come up with improvements. These can include take-profit levels, position sizing based on risk, limiting maximum drawdown, time-based exits, or volatility-based conditions.

Refer : StopLoss .ipynb VV_Stop_Loss.pdf

Metrics and Comparison :

Use your own backtesting engine or initial reference code provided during assignment 1 to test how your crafted strategy performs without risk management. You can compare these results by testing with the backtesting engine you have implemented integrating risk measures. The goal of assignment is to make you observe how risk control can improve stability, reduce large losses, and smoothen performance.

Note :

After incorporating risk management techniques, you might notice a slight drop in overall returns and that's okay. What matters more is how your strategy performs on key risk-adjusted metrics like drawdown and Sharpe ratio. However, to make sure that the trade-off is meaningful, the reduction in returns should be justified by an improvement in risk control. In many cases, a strategy with slightly lower returns but much better stability and lower drawdowns is far more practical for real-world trading.

Submission guidelines

- For the problem use the ticker symbol: 'BTC-USD'. Start date: '2018-01-01'(1st Jan 2018), end date: '2022-01- 12'(12th Jan 2018)
- In the .ipynb file RUN ALL THE CELLS BEFORE SUBMITTING and make sure your code is clean and modular. Try to keep your logic organized in functions, and if possible, add toggles to switch different exit or risk methods.
- Attach a document along with the submissions to explain the strategy and the risk management measure.
- Suggested methods: Trailing stop loss/take profits/ ATR based.

Alternatively: We would also like to see you incorporate position sizing along with your risk management. Let us know your views and results better through the report you will be making.

Contact any of us at any time, we are more than willing to help!