Simulating Financial Markets And Creating AI Powered Strategies

Ernest Zalewski & Nirius McDade

EXETER MATHS SCHOOL
FREEDOM FOR THINKERS

Introduction

Algorithmic trading, often known as algotrading, is the use of computer programs and algorithms to perform trades in a financial market. This has changed the niche of trading in modern financial markets by increasing execution efficiency, allowing for high-speed trading, and allowing for completely autonomous trades to take place. Over the last two decades, algorithmic trading has been accepted by most global hedge funds, proprietary trading firms, and retail investors alike. Our project researches this subject by investigating algotrading in the cryptocurrency markets with Python.

Decomposition & Setup

In order to make a trading strategy, we need to do the following:

1. Source historical price data for financial currency pairs
2. Install and configure a strategy backtesting framework
3. Create a script that evaluates whether to buy or sell depending on specific parameters
4. Evaluate the script's results and perform optimisations based on the output

We used a python package called freqtrade as a tool to create and test trading strategies. Configuring a basic environment to analyse strategies is as simple as typing a few commands into the terminal [1]:

freqtrade create-userdir –userdir user_data

freqtrade new-config –config user_data/config.json

freqtrade download-data –config config.json –timerange 20200101- –timeframes 1m 5m 15m 30m 4h 8h

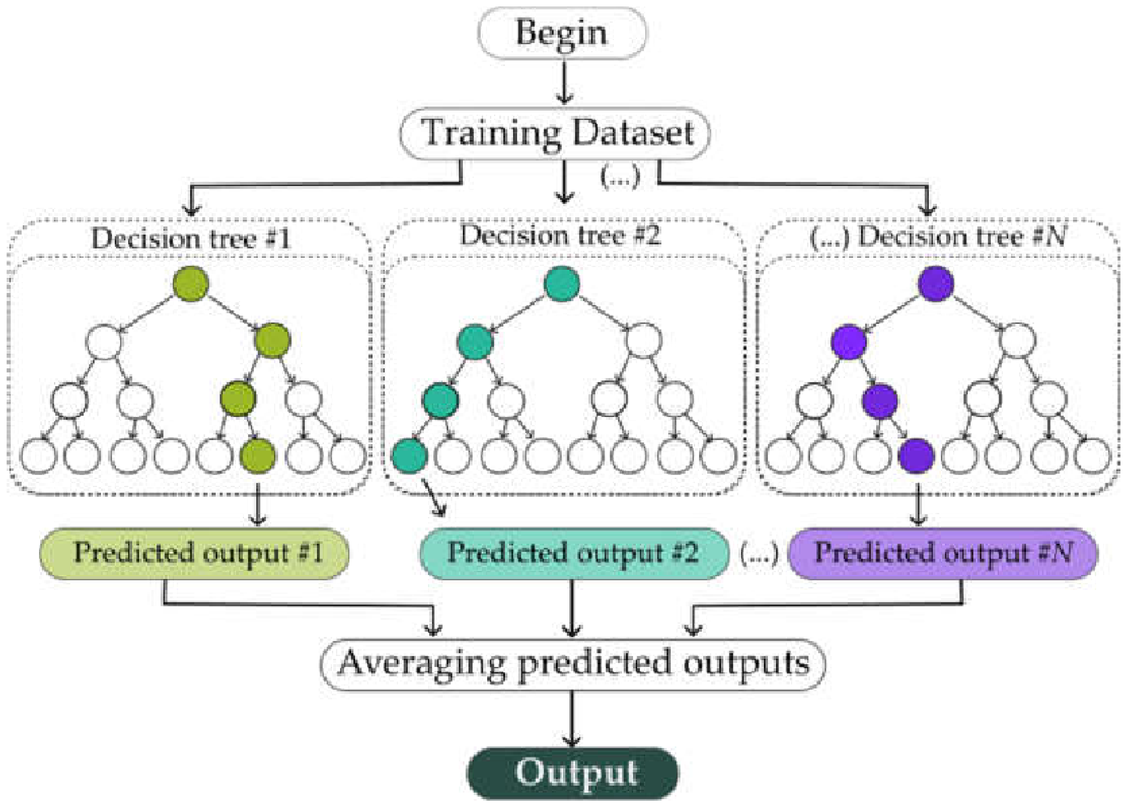
The first command creates a default directory for freqtrade. Then, a new configuration file that determines key factors about our strategies such as timeframe, what currency pairs to trade on, and maximum capital allocation is created. Finally, the last command retrieves all the pairs specified in our config file and collects price history for the indicated timeframes from January 1, 2020.

Methodology

Now we have a suitable environment to create a strategy. We started by creating a basic strategy that uses the RSI and MACD technical indicators [2]. The initial strategy was lacklustre since it performed worse than a simple buy and hold strategy. This meant that if we simply bought the currency pairs and did not sell it at all, we would have had more capital that if we used our strategy.

The screenshot above shows how the strategy performs suboptimally, creating multiple buy trades when the market is bearish. Therefore we need to optimise this strategy in order to increase its profitability and reduce losing trades.
We can optimise our strategy using hyperoptimisation by using the hyperopt function in freqtrade. This function uses a Bayesian search algorithm in order to find the best values for the parameters in our strategy. The search uses a machine learning regressor algorithm (ExtraTreesRegressor) to model the relationship between hyperparameters and the performance of the trading strategy. What this means is that the hyperopt function searches through the most probable values that minimise the amount of loss, for example, minimising the value of negative profit in our trading strategy.

Many other loss functions exist, and we primarily used the SharpeHyperOptLoss function, which minimises the loss calculated on trades relative to the standard deviation of the returns of our strategy [3]. An high level explanation of the regressor algorithm is below [4]



This optimisation is done by using the following command:

freqtrade hyperopt –config config.json –strategy sample_strategy –hyperopt-loss SharpeHyperOptLoss –timerange 20220101-

Results & Conclusions

After many hours of experimenting with different indicators and parameters and after hyperoptimisation, we settled on a final strategy. Below is a segment of our results:

Metric	Value
Starting balance	1000 USDT
Final balance	1084.174 USDT
Total profit %	8.42%
CAGR %	109.06%
Profit factor	8.12
Avg. daily profit %	0.21%
Max % of account underwater	0.59%
Market change	-22.32%

The chosen currency pairs shrunk by 22.32% in a given time range, but our strategy netted us a profit of 8.42% . This shows how we successfully created a strategy that outperforms the market.
However, the time range that we used for this backtest was only a few months long, which is relatively short. The strategy underperforms the market in most longer backtests, showing how the strategy is inconsistent. This means that we would need to change the strategy in order to keep a constant profit. We were also limited by our computational power, as hyperoptimisation requires a constant use of many CPU threads, often for several consecutive days without interruption. Despite these setbacks, our investigation has discovered the logic behind algorithmic trading, how to analyse a strategy, and how to customise and improve an algotrading strategy.

References

[1] https://docs.freqtrade.io/en/2024.11/configuration/
[2] https://docs.freqtrade.io/en/2024.11/strategy-101/
[3] https://docs.freqtrade.io/en/2024.11/hyperopt/
[4] https://www.researchgate.net/profile/Ligia-Silva-15/publication/372103562/figure/fi
diagram-of-the-ExtRa-Trees-algorithm-regression-process.png