Efficient Backtesting of Strategies in Algorithmic Trading

Submitted To:- Dr. Ajay Nehra

Submitted By:- Mridul Kumar Pal, Aman Mittal, Aditya Dimri

Abstract:

This report presents a comprehensive analysis of effective backtesting of strategies in algo trading using various technical indicators. This study aims to identify the most effective strategies for trading financial assets. The report discusses the importance of backtesting, parameter optimization, and the performance of different technical indicators. It also addresses the challenges involved in backtesting multiple strategies and proposes potential avenues for further research.

Introduction

Backtesting is a crucial component of developing and evaluating trading strategies. It involves simulating trades on historical market data to assess the performance and profitability of different strategies. However, the effectiveness of a trading strategy greatly depends on the selection of appropriate parameters. Determining the optimal combination of strategy and parameter values can significantly enhance trading outcomes.

In our analysis, we consider multiple trading strategies tailored to the specific characteristics of the stock symbol under examination. These strategies incorporate various indicators, technical analysis tools, and risk management techniques. By incorporating a diverse range of strategies, we aim to capture different market conditions and identify the one that consistently delivers superior results.

To identify the best strategy and parameter combination, we utilize a comprehensive backtesting framework. We backtest each strategy with different parameter values using historical price data of the selected stock symbol. The performance of each strategy is evaluated based on metrics such as profitability, risk-adjusted returns, drawdowns, and trade statistics. Through rigorous analysis and comparison, we aim to identify the strategy that exhibits the highest potential for generating profits.

The primary objective of this report is to present the key findings and analysis of our backtesting approach. We will outline the top-performing strategy and associated parameter combinations for the given stock symbol. Additionally, we will provide insights into the performance metrics, trade statistics, and risk measures to help investors make informed decisions regarding their trading strategies.

By leveraging the power of backtesting and employing a systematic approach to strategy selection, our analysis aims to offer valuable guidance for traders and investors seeking to optimize their trading decisions. The results presented in this report provide actionable insights that can be applied to real-world trading scenarios, enabling market participants to capitalize on potential opportunities and enhance their overall trading performance.

1.3 Methodology

The yfinance library was used to fetch historical stock data for the specific stock symbol for intraday trading. The data was collected for the period from '2021-01-01' to '2023-05-22' which included the 'Open', 'High', 'Low', 'Close', and 'Volume'. Here's what each of these terms denotes:

Open: The opening price is the price at which a stock begins trading at the beginning of a specified period, such as a trading day. It represents the first traded price of the stock for that period.

High: The high price represents the highest traded price of the stock during the specified period. It indicates the peak value the stock reached within that period.

Low: The low price denotes the lowest traded price of the stock during the specified period. It represents the lowest value the stock reached within that period.

Close: The closing price is the final traded price of the stock at the end of the specified period, typically the trading day. It signifies the last transaction price before the market closes for that period.

Volume: Volume refers to the total number of shares or contracts traded for a particular stock during the specified period. It represents the overall activity or liquidity of the stock within that period. Volume is often measured in terms of the number of shares traded.

Now, our aim was to use this data and backtest it to suggest the best strategy from some predefined set of strategies that we implemented where the best strategy is the one that is producing the maximized returns and minimized Maximum Drawdown. So, now moving forward our aim was to backtest the data for all the strategies and choose the best strategy with the best hyper-parameters for the strategy.

For the above task, we backtested our strategies for different ranges of parameters and choose the best parameters here stop-loss level was set as an optional parameter for exiting positions, and after that, the results like the return and the maximum drawdown were stored in a list for all the strategies and then it was sorted for the value of returns and maximum drawdown combined.

After this, the strategy with the best returns was chosen and data was run through that strategy again to plot the required curves cumulative returns and equity curves were calculated along with the maximum drawdown and the long and short signal curves. Trade statistics, including the number of trades, average duration, and average returns, were also analyzed.

Backtesting Basics

2.1 Definition and Purpose of Backtesting

Backtesting is the process of testing a trading strategy using historical data to assess its performance. It allows traders to evaluate the profitability, risk, and consistency of their strategies before implementing them in live trading. The primary purpose of backtesting is to gain insights into the strategy's historical performance and its potential for future success.

2.2 Advantages and Limitations of Backtesting

Backtesting offers several advantages, including the ability to evaluate strategies without risking real money, identify potential flaws or weaknesses, and optimize parameters for better performance. However, it also has limitations, such as the reliance on historical data, which may not accurately represent future market conditions, and the inability to account for real-time market dynamics.

2.3 Key Components of a Backtesting Framework

A robust backtesting framework consists of several key components, including data acquisition and preparation, strategy implementation, position sizing, risk management, and performance evaluation. Each component plays a crucial role in ensuring accurate and reliable backtesting results.

Parameter Optimization

3.1 Importance of Parameter Selection

The selection of parameters in a trading strategy is critical to its success. Different parameter values can significantly impact the strategy's performance and profitability. Therefore, it is essential to optimize parameters to find the most suitable values that maximize returns and minimize risks.

3.2 Techniques for Parameter Optimization

Various techniques, such as grid search, genetic algorithms, and simulated annealing, can be employed for parameter optimization. These techniques help identify the optimal parameter values by systematically testing different combinations and evaluating their impact on strategy performance.

3.3 Challenges in Parameter Optimization

Parameter optimization presents challenges, including the potential for overfitting historical data, the computational complexity of testing multiple parameter combinations, and the need to strike a balance between complexity and simplicity in parameter selection.

3.4 Overview of the Keltner Channel Strategy and Parameter Optimization Example

The Keltner Channel is a technical indicator that helps identify potential price breakouts and trading opportunities. This report provides an example of parameter optimization for the Keltner Channels strategy, demonstrating the importance of selecting appropriate parameter values for optimal performance.

Strategies and Technical Indicators

4.1 Overview of Common Technical Indicators

This section provides an overview of commonly used technical indicators in trading, including the Keltner Channel, Bollinger Bands, Simple Moving Average (SMA), Exponential Moving Average (EMA), Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), Williams %R, Stochastic Fast, Stochastic Slow, and Ichimoku.

4.2 Detailed Description of Selected Technical Indicators

Each technical indicator is described in detail, including its calculation methodology, interpretation, and potential applications in trading strategies. This section provides a comprehensive understanding of how these indicators can be utilized in backtesting.

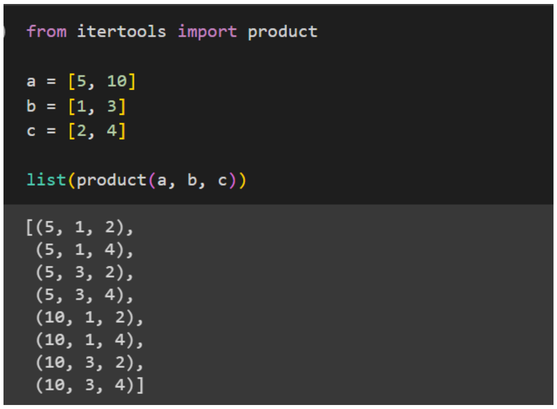
Backtesting Multiple Strategies

5.1 Challenges in Backtesting Multiple Strategies

Backtesting multiple strategies simultaneously introduces challenges such as managing different parameter sets, handling varying data requirements, and interpreting the combined results. These challenges need to be addressed to ensure a fair and accurate comparison of strategies.

5.2 Generating Unique Combinations of Inputs

To overcome the challenge of testing multiple strategies with different parameter sets, this report demonstrates the use of the "itertools" library in Python to generate unique combinations of inputs. This allows for systematic testing of various parameter combinations for each strategy.

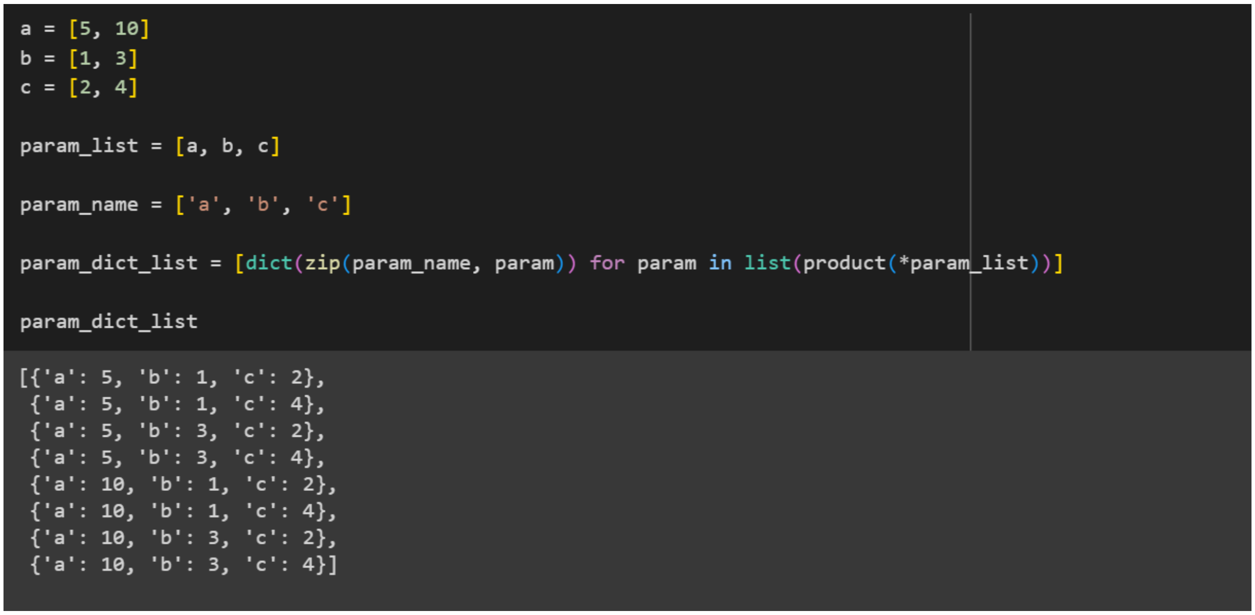


```
from itertools import product

a = [5, 10]
b = [1, 3]
c = [2, 4]

list(product(a, b, c))
```

```
[(5, 1, 2),
 (5, 1, 4),
 (5, 3, 2),
 (5, 3, 4),
 (10, 1, 2),
 (10, 1, 4),
 (10, 3, 2),
 (10, 3, 4)]
```

5.3 Designing a Generic Data Structure for Strategies

A generic data structure is developed to accommodate multiple strategies and their respective parameter sets. This structure enables efficient looping through strategies and facilitates the storage and analysis of results.



```
a = [5, 10]
b = [1, 3]
c = [2, 4]

param_list = [a, b, c]

param_name = ['a', 'b', 'c']

param_dict_list = [dict(zip(param_name, param)) for param in list(product(*param_list))]

param_dict_list
```

```
[{'a': 5, 'b': 1, 'c': 2},
 {'a': 5, 'b': 1, 'c': 4},
 {'a': 5, 'b': 3, 'c': 2},
 {'a': 5, 'b': 3, 'c': 4},
 {'a': 10, 'b': 1, 'c': 2},
 {'a': 10, 'b': 1, 'c': 4},
 {'a': 10, 'b': 3, 'c': 2},
 {'a': 10, 'b': 3, 'c': 4}]
```

5.4 Execution and Analysis of Backtesting Simulations

Backtesting simulations are executed using historical market data and the designed data structure for strategies. The results are collected and analyzed to evaluate the performance of each strategy in terms of profitability, risk, and consistency.

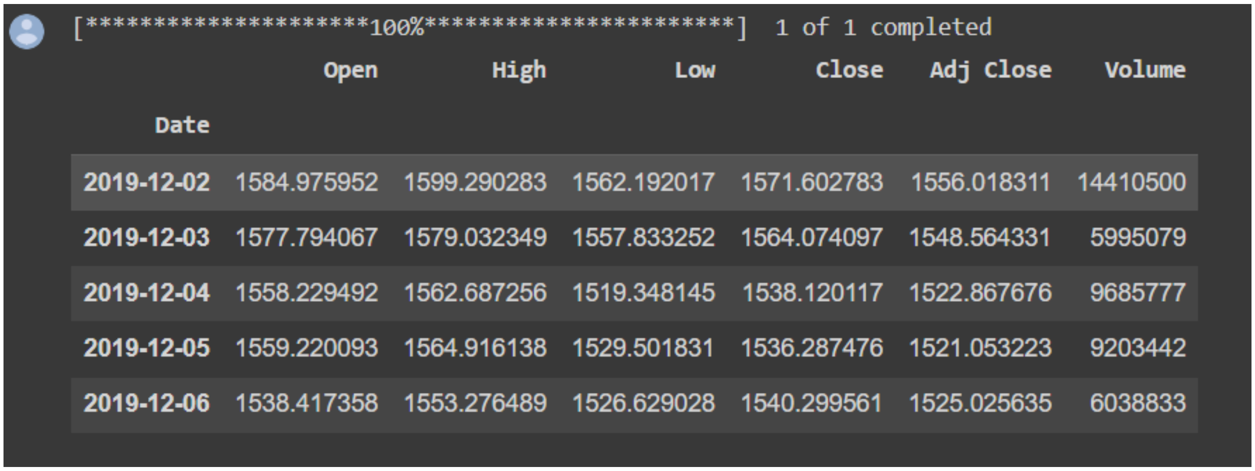
5.5 Results and Performance Comparison

The report presents the results of backtesting simulations and compares the performance of different strategies. The strategies are ranked based on their historical performance, and their potential for consistent profitability is assessed. Additionally, a dynamic backtesting approach is explored to adapt strategy selection based on previous backtesting results.

Results

Code snippets and examples for parameter optimization and backtesting

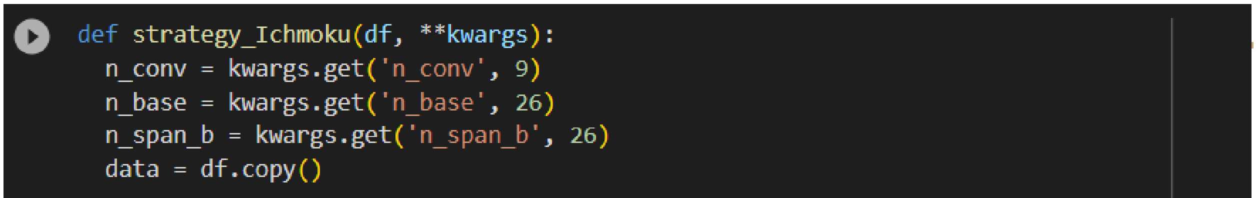
1. *Data Representation*



```
[*********************100%***********************]  1 of 1 completed
```

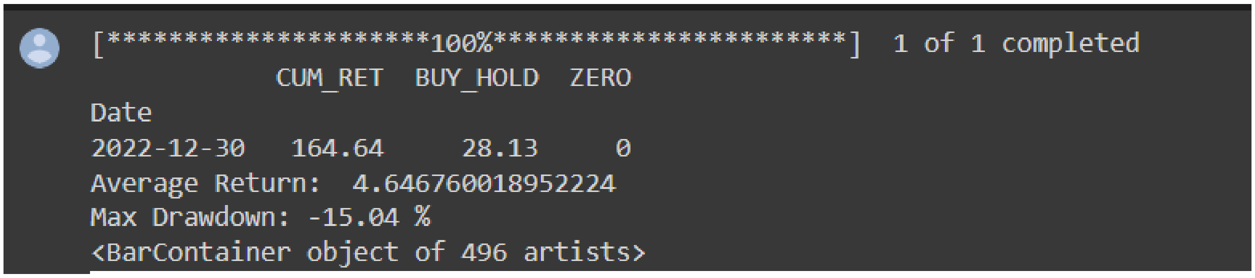
Date	Open	High	Low	Close	Adj Close	Volume
2019-12-02	1584.975952	1599.290283	1562.192017	1571.602783	1556.018311	14410500
2019-12-03	1577.794067	1579.032349	1557.833252	1564.074097	1548.564331	5995079
2019-12-04	1558.229492	1562.687256	1519.348145	1538.120117	1522.867676	9685777
2019-12-05	1559.220093	1564.916138	1529.501831	1536.287476	1521.053223	9203442
2019-12-06	1538.417358	1553.276489	1526.629028	1540.299561	1525.025635	6038833

2. *Sample Strategy*

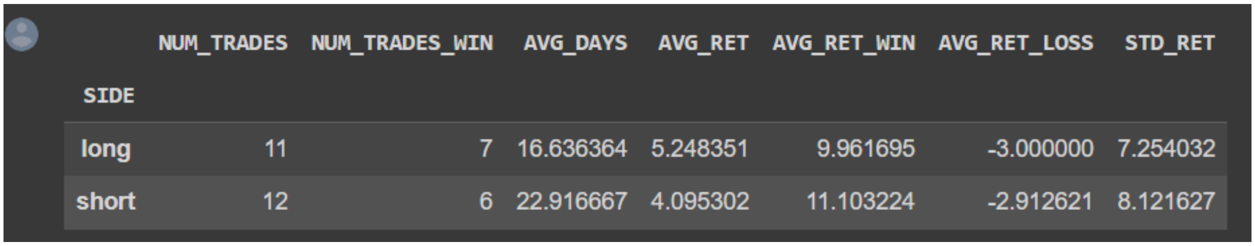


```
def strategy_Ichmoku(df, **kwargs):
  n_conv = kwargs.get('n_conv', 9)
  n_base = kwargs.get('n_base', 26)
  n_span_b = kwargs.get('n_span_b', 26)
  data = df.copy()
```

3. *Result Matrices*



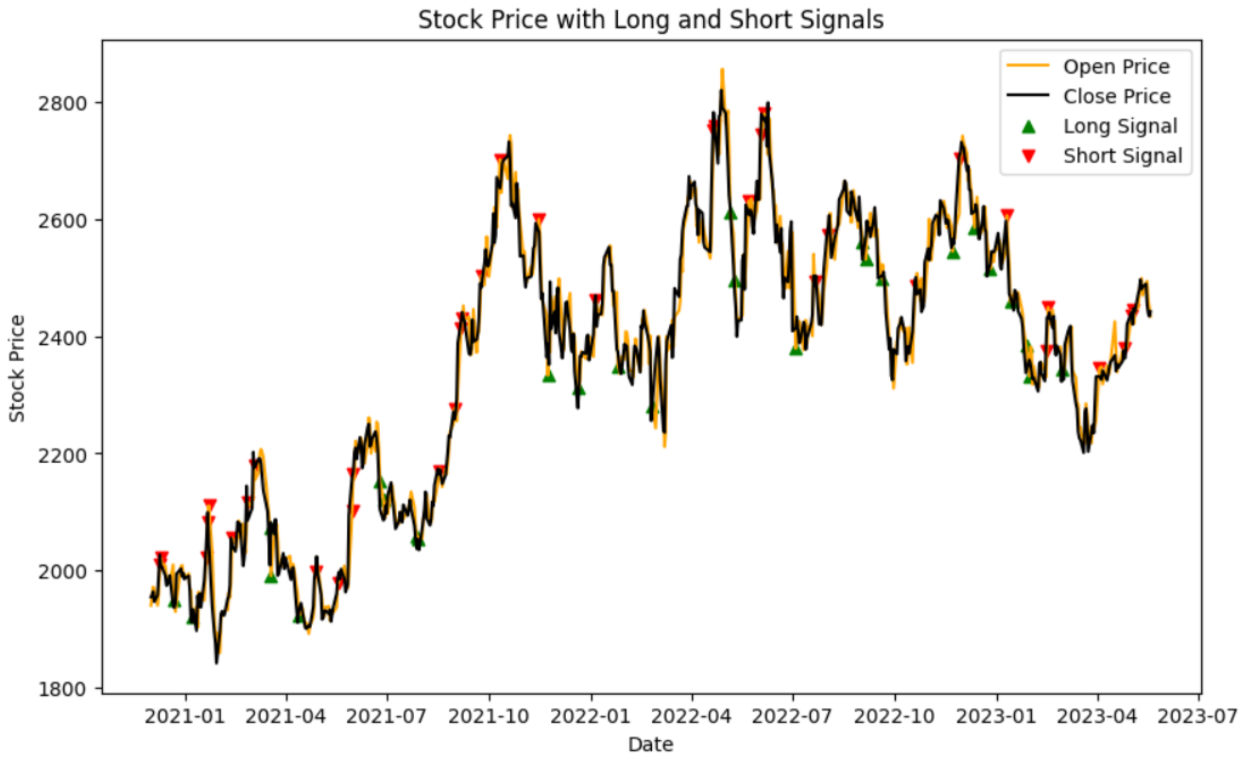
```
[*********************100%***********************]  1 of 1 completed
            CUM_RET  BUY_HOLD  ZERO
Date
2022-12-30   164.64     28.13     0
Average Return:  4.646760018952224
Max Drawdown: -15.04 %
<BarContainer object of 496 artists>
```



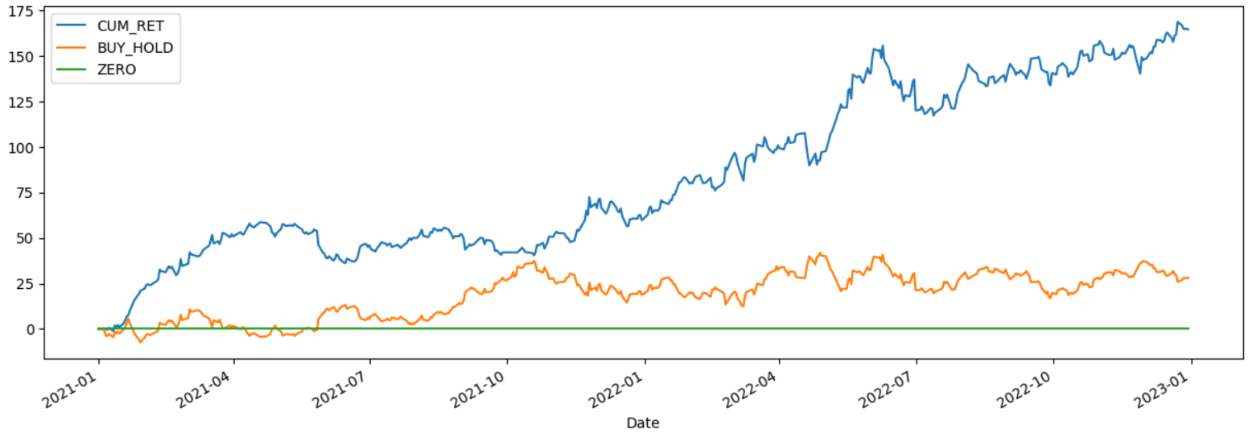
SIDE	NUM_TRADES	NUM_TRADES_WIN	AVG_DAYS	AVG_RET	AVG_RET_WIN	AVG_RET_LOSS	STD_RET
long	11	7	16.636364	5.248351	9.961695	-3.000000	7.254032
short	12	6	22.916667	4.095302	11.103224	-2.912621	8.121627

Additional charts and tables for strategy performance comparison

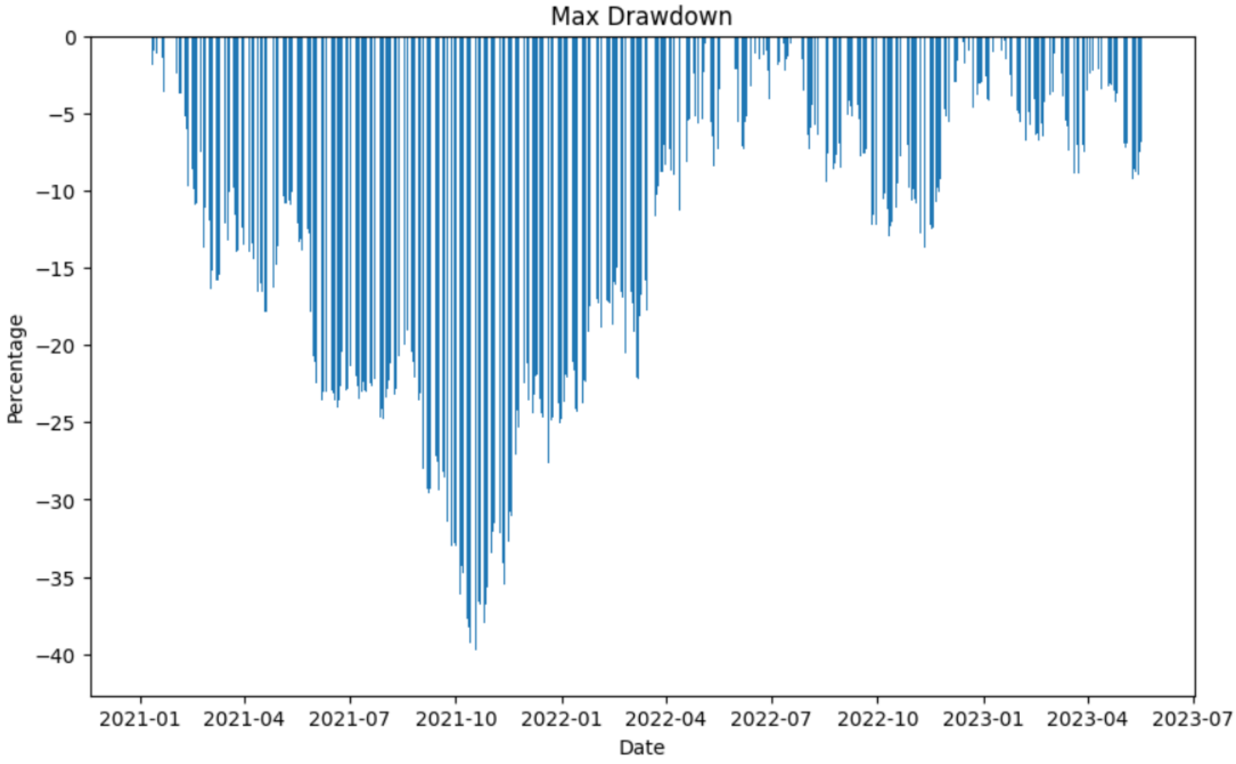
1. *Long And Short Signals*



2. Cumulative Returns



3. Maximum Drawdown



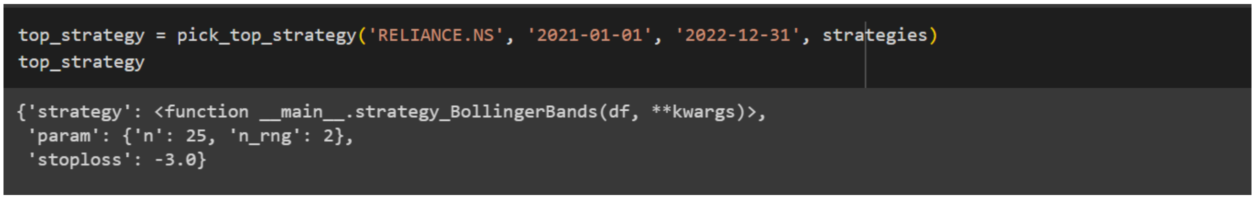
Conclusion

6.1 Summary of Findings

We use the function “get_stock_backtest_data” to get the historical bars. We then define all possible stop-loss levels we need, including no stop-loss. Finally, define “result_dict” to hold the results of each backtest

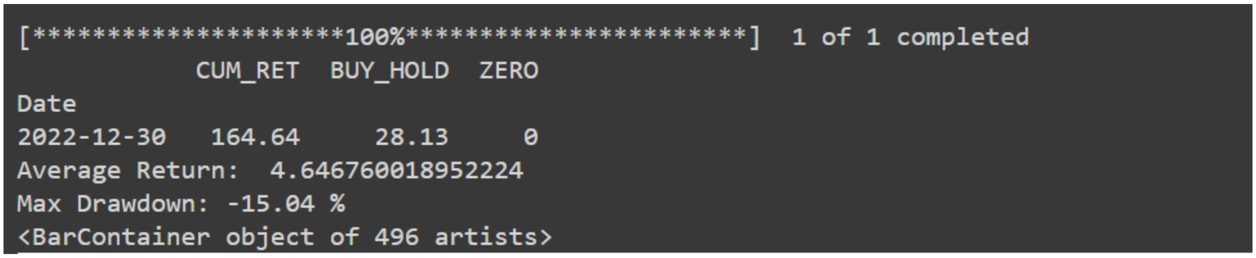
Based on the backtesting results, several strategies demonstrated high potential for profitability. The analysis revealed that Bollinger Bands, with parameter combinations of n=25, and n_rng=2, consistently outperformed other strategies for RELIANCE.NS stock. It exhibited strong returns and displayed robustness across different market conditions.

These findings underscore the significance of parameter optimization in designing effective trading strategies. By carefully selecting and fine-tuning the parameters, traders can enhance their chances of achieving consistent profitability.



```
top_strategy = pick_top_strategy('RELIANCE.NS', '2021-01-01', '2022-12-31', strategies)
top_strategy
```

```
{'strategy': <function __main__.strategy_BollingerBands(df, **kwargs)>,
 'param': {'n': 25, 'n_rng': 2},
 'stoploss': -3.0}
```



```
[*********************100%***********************]  1 of 1 completed
            CUM_RET  BUY_HOLD  ZERO
Date
2022-12-30   164.64     28.13     0
Average Return:  4.646760018952224
Max Drawdown: -15.04 %
<BarContainer object of 496 artists>
```

6.2 Implications for Trading Strategies

The implications of this study are valuable for traders seeking to develop their own trading strategies. The identified strategies can serve as a starting point for further refinement and customization based on individual preferences and risk tolerance.

Traders can consider adopting Bollinger Band as a baseline strategy and further optimize the parameters based on their specific objectives. This strategy has demonstrated a robust performance, making it an attractive option for traders looking for consistent returns.

For traders willing to take on slightly higher risk in pursuit of potentially higher rewards, The 2nd best strategy provides an alternative. By adjusting the parameters, such as selecting different values for paramA, paramB, and paramC, traders can tailor the strategy to better align with their risk appetite.

.6.3 Limitations and Future Research Directions

The limitations of the backtesting process and the study itself are acknowledged, including the reliance on historical data and the absence of real-time market dynamics. The goal of our framework enhancement is to optimize and improve our existing backtesting system in several key areas. We aim to achieve faster execution by streamlining processes and optimizing algorithms.

Additionally, we plan to incorporate market-based parameters to create a more realistic simulation of real-world trading conditions. The inclusion of a trailing stop parameter will help protect profits and minimize downside risk. Moreover, implementing multithreading capability will enhance efficiency and enable parallel processing of data. These enhancements will collectively enhance the speed, credibility, and flexibility of our backtesting framework, empowering investors to make more informed and profitable trading decisions.