

Kivi Assignment

Samanyu Mahajan

Contents

1	Testing	1
2	Strategy	2
3	Indicators	2
3.1	Initial Attempt	2
3.2	Bollinger Bands	2
3.3	RSI	2
3.4	+DI	2
3.5	Why this works	2
3.6	Threshold Selection	3
4	Entering trades	3
4.1	Long condition:	3
4.2	Short Condition:	3
5	Updating Indicators	3
6	PNL Over the years	4
6.1	2018	4
6.2	2019	5
6.3	2020	5
6.4	2021	5
6.5	2022	5
6.6	2023	6
7	Execution	7
8	Logging	7
9	Metrics	7

1 Testing

lot size is taken as 1. can be changed in config.py. A simulation for a year takes about 25 sec to run. To test on out of sample data, make changes in the config namely:

- DATA_FILE: path of data
- START_DATE: change to 20240101
- END_DATE: change to 20241231

and run using ./run.sh.

2 Strategy

Base Strategy handles the logging etc. DGLongShort derived from BaseStrategy has the actual logic of the strategy and works as follows. Based on Indicators we identify long and short conditions. Suppose long condition is met then we place an aggressive order. We also calculate the **Stop loss** and the Target. For a long position if the price falls below the stop loss, we exit. If it rises above the target, we lock in our gains and exit. Similar but reverse logic for a short position. Finally at the end of holding period (4 days) we exit all open positions.

3 Indicators

3.1 Initial Attempt

Tried: Only RSI and SMA

Initially, I used just RSI and a simple moving average (SMA) to capture general price direction.

Problem: It often led to false signals Why it failed: RSI is reactive but not sufficient alone; SMA smoothed too much. Finally I settled on the following:

3.2 Bollinger Bands

- Middle Band: SMA(14)
- Upper Band: SMA + 2 Standard Deviation
- Lower Band: SMA - 2 S.D

when price touches the Lower Band it indicates oversold condition. Price has hit the support and is likely to bounce back thus it is a buying signal. On the other hand if the price touches the upper band it is an overbought condition. Price has hit the resistance and is likely to drop hence it is a selling signal. Further a wider band (difference between upper and lower bands) implies higher volatility. A lower band implies sideways movement and we don't want to trade then. So we check if the band is greater than a threshold before entering into any position.

3.3 RSI

$$RSI = 100 - \frac{100}{1 + RS} \quad (1)$$

where

$$RS = \frac{\text{AverageGainoverNperiods}}{\text{AverageLossoverNperiods}} \quad (2)$$

where $N = 14$. When $RSI > 70$ then it is an overbought situation. When $RSI < 30$ it is an oversold situation. We use RSI Rate of change. When the rate of change is negative and large it means that RSI has dropped significantly, which indicates momentum to the downside or oversold conditions which is a buying signal. Similarly when Rate of change of RSI is large positive it indicates upward momentum and likely overbought situation so it is a sell signal.

3.4 +-DI

+DI is a smoothed version of +DM which indicates the strength of upward momentum, whereas -DI indicates the strength of downward momentum. Consider the rate of change of -DI, if it is large and negative this means that -DI is decreasing at a fast rate which means that the bearish trend is fading away which is a buying signal. Similarly when rate of change of +DI is large positive this indicates strong bullish movement however this may imply an overbought position and hence considered as a selling signal.

3.5 Why this works

- RSI gives short-term momentum.
- DI+/DI- offer directional strength.
- Bollinger Bands introduce a volatility filter.
- SMA is helpful for smoothing

Together, these indicators balance each other.

3.6 Threshold Selection

- a significant change in RSI (`rsi_rc`) might be $\pm 5-10\%$, which indicates a clear momentum buildup or shift.
- `plus_di_rc` and `minus_di_rc`: 5% is enough change to signify a meaningful move but not too sensitive to noise.
- Bollinger Band width (`band_diff`) was crucial in filtering out trades during low-volatility periods (> 0.4)

$RSI_{RC} < -x$ (long condition). $x = 5$ is chosen because for a large value of x like 50 as the condition now is tighter, fewer number of trades are executed, PNL is lower so choosing a single digit value seems reasonable. Similar reasoning goes for $+DI$. Running sim for the year 2020.

x	PNL	Total trades
50	22931	236
5	52392	431

Bollinger Band width = x . For a high value of x like 10 the constraint is too tight, no trades are executed. Thus it is reasonable to have $x < 1$. We chose $x = 0.4$.

x	PNL	Total trades
10	0	0
0.4	52392	431

4 Entering trades

4.1 Long condition:

- `self.candle['low'] < self.BBANDS.lowerband`: price is lower than low band indicates oversold condition. price is likely to rise.
- `self.rsi_rc <= -5`: RSI is decreasing indicates hitting support. price is likely to rise
- `self.minus_di_rc < -5`: bearish trend is fading away. price is likely to rise
- `self.band_diff > 0.4`: market is volatile

4.2 Short Condition:

- `self.candle['high'] > self.BBANDS.upperband`: price is higher than high band, indicates overbought condition, likely to fall now.
- `self.rsi_rc > 5`: RSI is increading indicates hitting resistance. likely to fall
- `self.plus_di_rc > 5`: strong bullish trend. likely to reverse. price likely to fall
- `self.band_diff > 0.4`: market is volatile

5 Updating Indicators

Running sim for year 2020. The time frame after which indicators are updated is varies from 15 minutes to 60 minutes in multiples of 5

We obtain a good PNL for 60 minutes and it takes less time to run so we chose this as our 'update minutes'.

$Time(minutes)$	PNL
15	24881
20	-11588
25	55658
30	46748
35	10761
40	31523
45	61412
50	51700
55	48648
60	52392

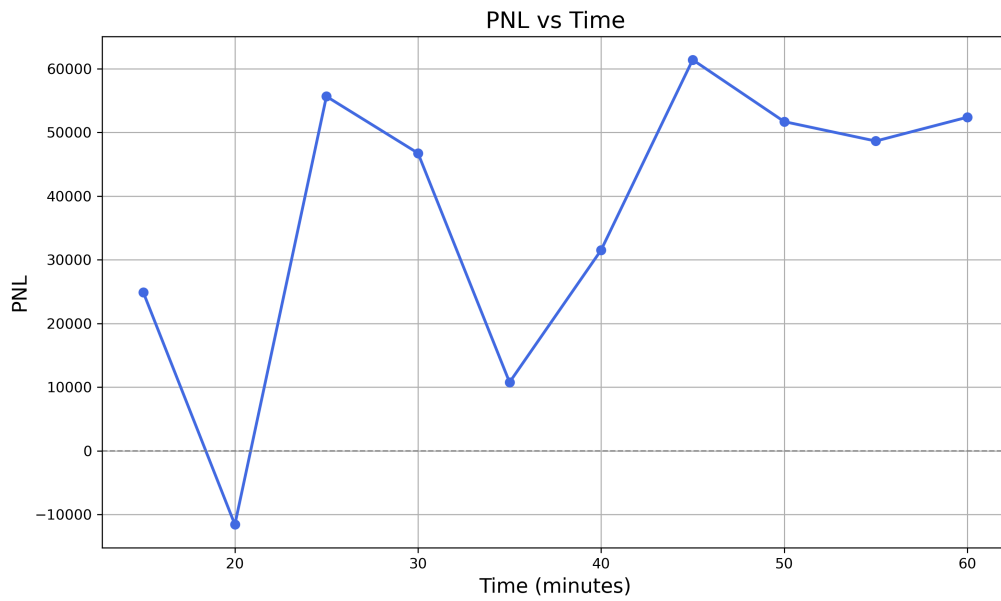


Figure 1: PNL vs Time plot

6 PNL Over the years

for update minutes = 60.

Year	<i>PNL</i>
2015	-8275
2016	-4208
2017	7139
2018	-3621
2019	5276
2020	52392
2021	6686
2022	29135
2023	10528

6.1 2018

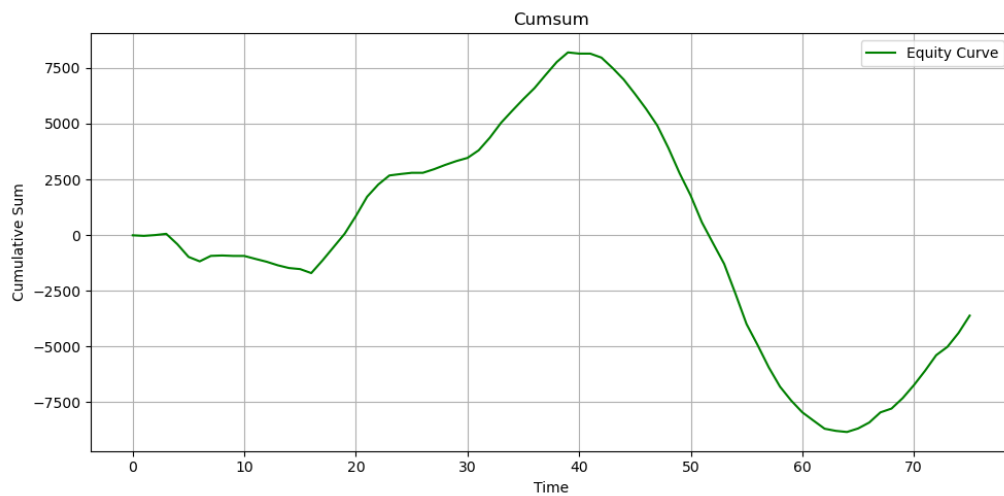


Figure 2: 2018

Total PNL	-3621
Max Drawdown	-32.64

6.2 2019

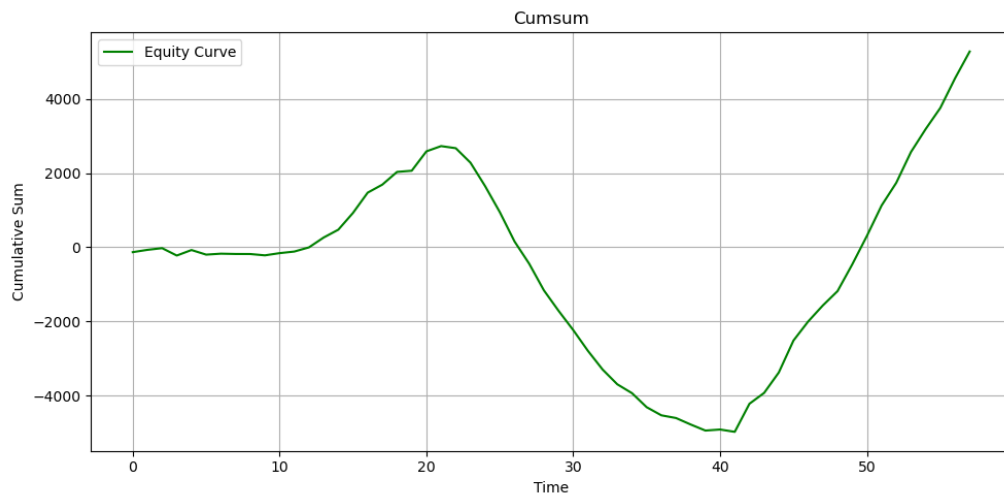


Figure 3: 2019

Total PNL	5276
Max Drawdown	-2.82

6.3 2020

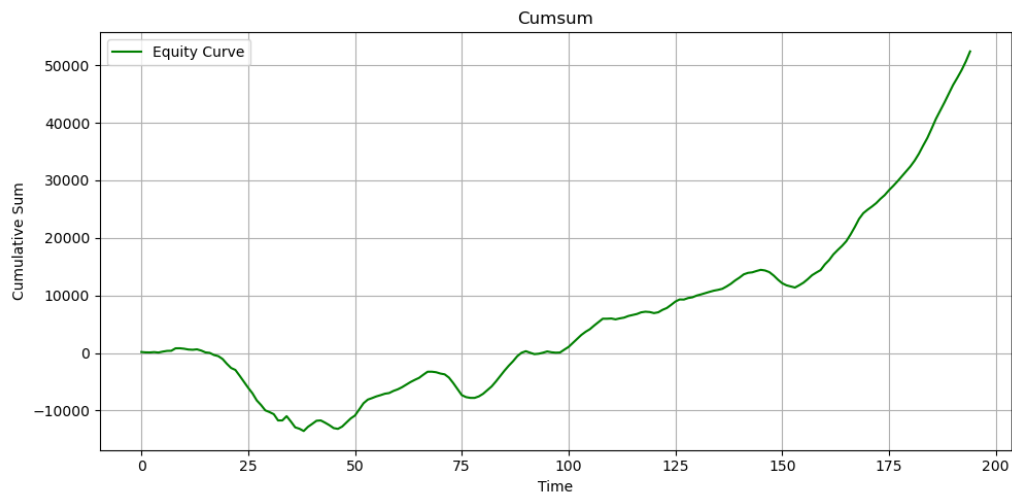


Figure 4: 2020

Total PNL	52392
Max Drawdown	-17.39

6.4 2021

Total PNL	6686
Max Drawdown	-1.83

6.5 2022

Total PNL	29135
Max Drawdown	-5.53

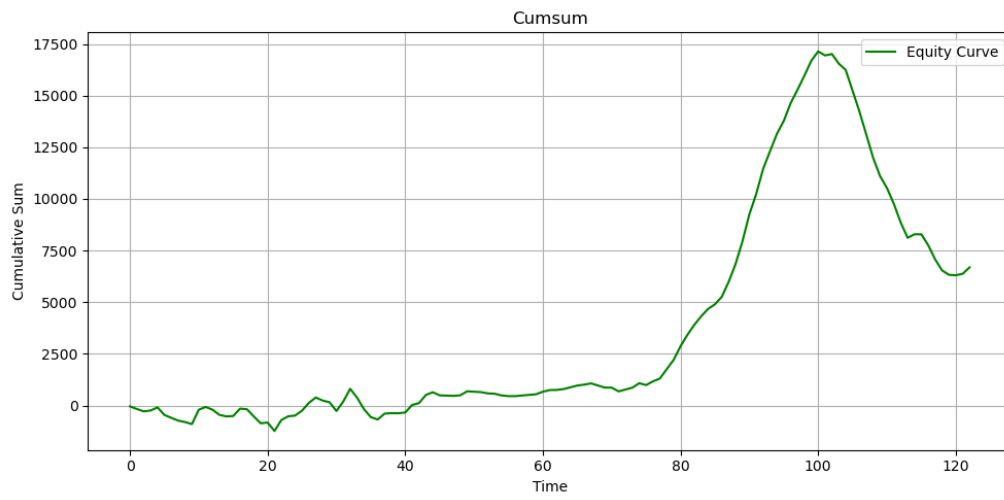


Figure 5: 2021

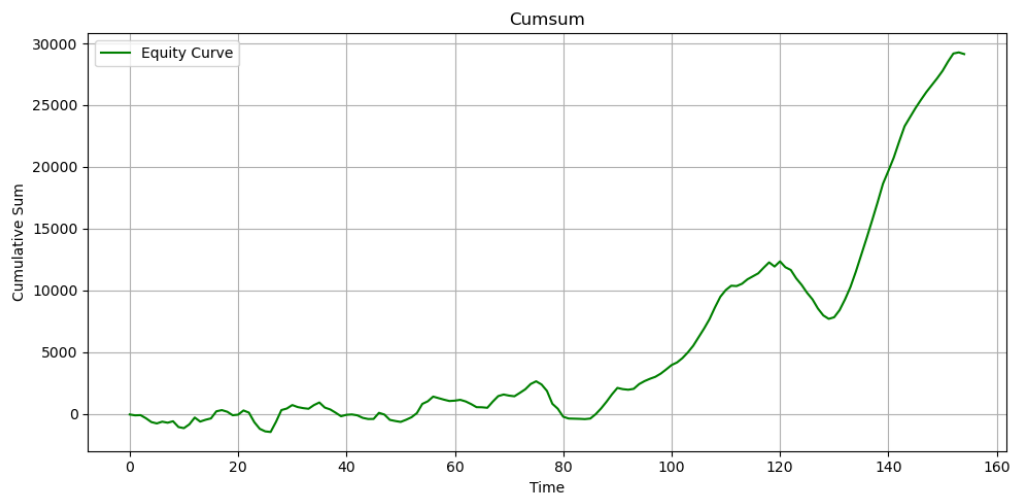


Figure 6: 2022

6.6 2023

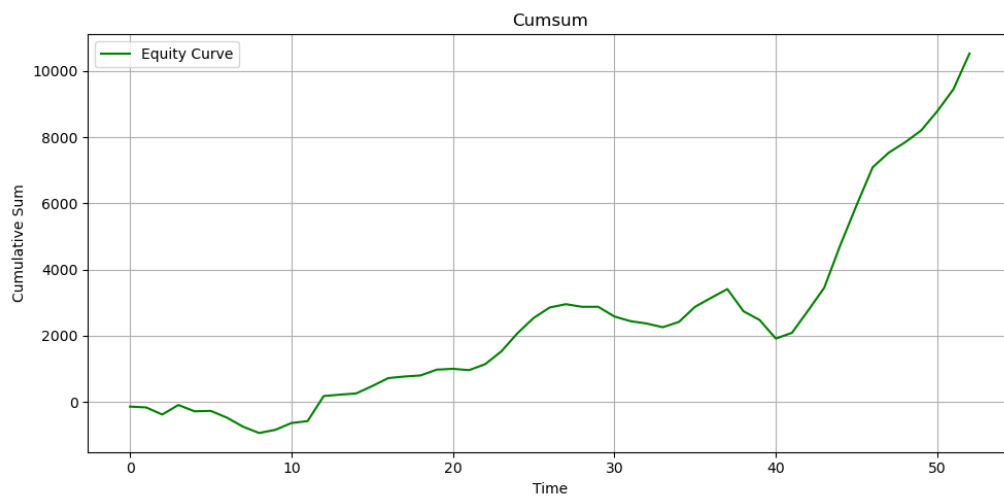


Figure 7: 2023

Total PNL	10528
Max Drawdown	-0.43

7 Execution

We run a simulation. Datastore object fetches the packets between start and end dates and then each packet is given to the exchange and then to the strategy sequentially. The strategy reads the packet and does stuff (placing order/updating indicators). Orders are stored in a list. The exchange on receiving a packet fills the orders which would have been filled, logs the order and then update the strategy. The strategy then keeps a record of this using class "Position". At the end of simulation, the pnl is calculated and displayed in the logs.

8 Logging

logs are made in folder Code/logs/start_date/update_minutes where start_date and update_minutes are as in config.py. We maintain the following logs:

- stdout.log: the final results (metrics) of the simulation
- stats.csv: same thing as above in csv format
- order_details: informative details of orders like long position met at what time and price, squaring off etc.
- orders.csv: log of placed but pending orders
- fill_orders.csv: log of filled orders.

The first three logs are maintained in base strategy and the last two in exchange(executor.py)

9 Metrics

For each simulation, the following are recorded.

- Total PNL
- Total Orders
- Total volume traded
- Winning Trades
- PNL turnover ratio in bps
- Sharpe Annually
- Max Drawdown
- plot of cumsum