COMP0051 Algorithmic Trading

Coursework - 2

22092172

UCL

April 6, 2023

0.1 Introduction

This report explains trading strategies using leverage on S&P500 ETF (called SPDR or Spider, ticker SPY Equity[1]) end-of-day prices for the period of time between 1 Jan 2014 to 31 December 2019. The three trading strategies used here are - Autoregression mean reversion strategy, Simple moving average crossover strategy, and Moving average mean reversion strategy. The Methodology section explains the data preprocessing, implementation of trading strategies, and the performance indicators used in this report. The Results section shows the results extracted from the three strategies in the form of plots and performance indicator values. The project's conclusions are discussed in the Discussion part, and the Bibliography section lists the sources.

0.2 Methodology

0.2.1 Data

The SPDR end-of-day prices are downloaded using `yfinance` for the period of time between 1 Jan 2014 to 31 December 2019. The Effective Fed Funds Rate (EFFR Index) is downloaded using `full_fred` module in Python for the same time period. This risk-free rate contains missing values which are filled by the forward fill method where any null values are replaced with the last non-null value in the column. The risk-free rate values from EFFR are adjusted to the daily rate r_t^f by dividing by 252 * 100 and saved in the `daily_rate` column.

The daily excess return per unit of SPDR asset prices i.e. 'Close' prices is calculated using the following formula and saved in the `excess_return` column:

$$r_t^e = \frac{\Delta p_t}{p_t} - r_t^f$$

The daily returns of the SPDR Close prices are calculated using `pct_change()` function and stored in the `simple_return` column of the data which is plotted.

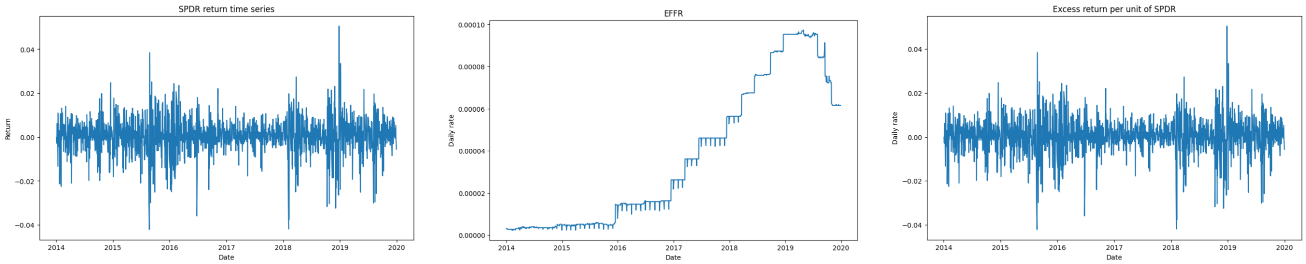
The first five entries of the combined data of SPDR and EFFR are as follows:

date	Open	High	Low	Close	Adj Close	Volume	value	daily_rate	simple_return	excess_return
2014-01-02	183.979996	184.070007	182.479996	182.919998	154.400269	119636900	0.08	0.000003	NaN	NaN
2014-01-03	183.229996	183.600006	182.630005	182.889999	154.374878	81390600	0.08	0.000003	-0.000164	-0.000167
2014-01-06	183.490005	183.559998	182.080002	182.360001	153.927536	108028200	0.08	0.000003	-0.002898	-0.002901
2014-01-07	183.089996	183.789993	182.949997	183.479996	154.872955	86144200	0.07	0.000003	0.006142	0.006139
2014-01-08	183.449997	183.830002	182.889999	183.520004	154.906677	96582300	0.07	0.000003	0.000218	0.000215

Figure 1: Combined time series

Figures 2a, 2b, and 2c show the plots of SPDR return time series i.e. `simple_return`, EFFR i.e. `daily_rate` and excess return per unit of SPDR i.e. `excess_return` respectively built using `matplotlib` library of Python.

[1]https://finance.yahoo.com/quote/SPY/



(a) SPDR return time series (b) EFFR daily rates (c) Excess return per unit SPDR

Figure 2: Plots of SPDR and EFFR

The combined data time series is split into two sets: the training set and the test set. 70% of the dataset is used for training and the remaining 30% is used for testing.

0.2.2 Trading Strategies

In a leveraged strategy, the (leveraged) book size is the available capital times the leverage amount. By a leveraged strategy we mean a sequence ${\theta_t}^T = 1$ of dollar values of SPDR which can be long or short such that

$$|\theta_t| \leq V_t L$$

where V_t is the total value of the holdings, and L is the leverage. The initial price is set to $V0 = 200000$ and a Leverage $L = 5$. The three trading strategies used for this time series data are as follows:

1. **Autoregressive Mean Reversion strategy**: This strategy uses autoregression to predict the future price of a stock, comparing it to the current price to determine buy or sell signals. A buy signal is considered when the expected price is greater than the current price, and a sell signal is when the predicted price is lower than the current price. It uses a rolling window of past prices to train the autoregression model, with a window size of 30. The function `ar_mean_reversion` initializes variables for cash, units, and portfolio value, and iterates through historical Close prices. If the predicted price is greater than the current price, the code buys the security using available cash; if the predicted price is less than the current price, the code sells and receives cash in return. The `AutoReg` function is used from the `statsmodels.tsa.ar_model` module in Python. The *lags* parameter of the `AutoReg` function is set to $min(time\_window - 1, 5)$, which means that the model will include a maximum of 5 lags, or as many lags, as are available in the data up to 29. The strategy is evaluated using the PnL generated from the trades executed according to the Position vector.

2. **Simple Moving Average (SMA) crossover strategy**: This strategy involves calculating two moving averages with different time periods, a shorter SMA (30 periods) and a longer SMA (70 periods), and then comparing them to generate buy and sell signals. When the shorter-term moving average crosses above the longer-term moving average, it buys, and when it crosses below the latter, it sells. The simple moving average for n period is calculated as follows:

$$SMA_n = \frac{\sum_{i=1}^{n} Prices_i}{n}$$

The `sma_crossover` function is an implementation of the SMA crossover strategy in Python that uses the Close prices to calculate the two moving averages, and then generates buy and sell signals based on their relationship. The code keeps track of the cash, units, and portfolio value, and updates them based on the signals generated by the strategy. It also calculates the profit and loss (PnL) generated from the trades executed according to the Position vector.

3. **Moving average mean reversion strategy**: This strategy revolves around the assumption that the stock prices have a tendency to return to their historical mean. The $mean\_reversion$ function in Python calculates a moving average of the stock price over a fixed time window (30 periods), and if the moving average is equal to the Close price for that day, the portfolio does not make any trades, and the values of the portfolio variables are updated based on the previous day's values. If the moving average is greater than the price for that day, the portfolio sells its holdings and converts them to cash. Conversely, if the moving average is less than the price of the asset for that day, the portfolio uses its cash to buy.

The turnover of all the strategies is calculated over dollars which is given by :

$$Turnover_{dollar} = \sum_{0}^{T} |\theta_t|$$

as well as in the number of units traded over time:

$$Turnover_{units} = \sum_{0}^{T} |\frac{\theta_{t+1}}{p_{t+1}} - \frac{\theta_t}{p_t}|$$

The turnover is calculated on both the train and test sets and the values are reported in the Results section.

The PnL series for each of the strategies where it is assumed that the unused capital will be put in a money-market and grow at the same risk-free rate, i.e., the value of your account changes by the trading PnL (ΔV) and the change in the growth of the money-market capital account (ΔV_{cap}) is created using the following equation:

$$V_{t+1}^{total} - V_t^{total} = \Delta V_t^{total} = \Delta V_t + \Delta V_t^{cap} = (\frac{\Delta p_t}{p_t} - r_t^f) * \theta_t + (V_t^{total} - M_t) r_t^f$$

where V_{t+1}^{total} and V_t^{total} are the total values of the trading account at time $t+1$ and t, respectively. ΔV_t^{total} is the change in the total value of the account, which is the sum of the trading PnL ΔV_t and the change in the growth of the money-market capital account ΔV_t^{cap}, given by the product of the risk-free rate, r_t^f, and the unused capital in the account, $(V_t^{total} - M_t)$. V_t is the trading PnL, which is the product of the excess return of the asset and the position of the asset, θ_t. M_t is the total margin used, which is $|\theta_t|/L$. The function then calculates the total value of the account at each trading day, $total\_val[t]$, which is the sum of the previous total value of the account, $total\_val[t-1]$, and the change in the total value of the account, $V_{total}[t]/L$. Thus, the series function returns three arrays: V (trading PnL), V_{cap} (money-market PnL), and V_{total} (total PnL) for each of the strategies on Train and Test data.

0.2.3 Performance indicators

The four performance indicators used in this project are explained below. The results of the strategies on these indicators are mentioned in the Results section:

1. **Sharpe Ratio** : Ratio of the mean of daily excess returns PnL from the strategy to the standard deviation of the PnL. It assumes that the standard deviation of the returns distribution provides the full description of risk.

2. **Sortino Ratio** : Ratio of the mean of daily excess returns PnL from the strategy to the downside deviation of the PnL by selecting only the negative returns i.e. `df['pnl'] < 0`.

3. **Maximum Drawdown** : Difference between a peak in the investment value and the subsequent trough, divided by the peak value. It measures the maximum percentage decline from a portfolio's highest value to its lowest value over a certain period of time.

4. **Calmar Ratio** : Ratio of the average daily excess returns PnL from the strategy to the maximum drawdown over the same period of time. It is similar to Sharpe Ratio as it is a risk-adjusted measure of performance. but measures the maximum cumulative loss from a peak to the following bottom.

0.3 Results

The following plots show the Position of the strategies θ_t with the upper bound V_tL and lower bound $-V_tL$ for the three strategies on Train and Test Data:

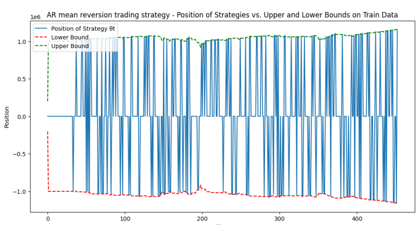


Figure 3: AR position vs bounds on train data

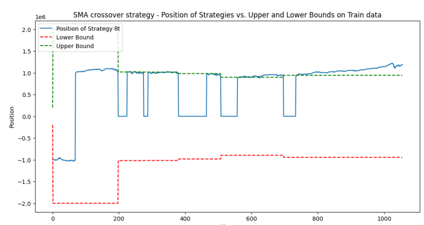


Figure 4: SMA crossover position vs bounds on train data

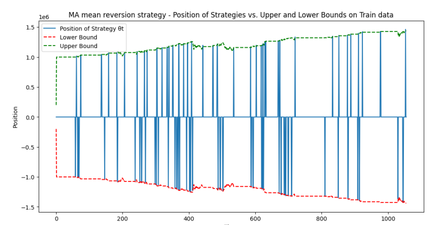


Figure 5: Mean reversion position vs bounds on train data

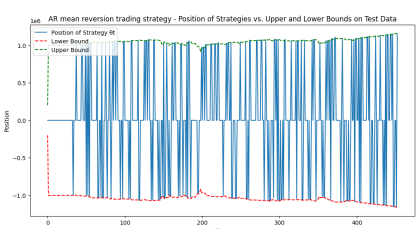


Figure 6: AR position vs bounds on test data

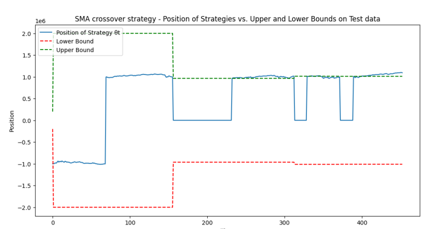


Figure 7: SMA crossover position vs bounds on test data

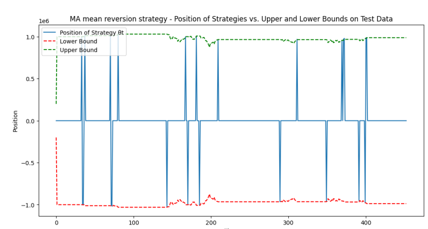


Figure 8: Mean reversion position vs bounds on test data

The Turnover in dollars and number of units over time for the Train and Test set are given in Table 1 and Table 2.

Strategy	Train data	Test data
Autoregression mean reversion	740246814.15	297418038.06
Simple moving average crossover	16020840.13	9935148.57
Moving average mean reversion	196681517.35	39673245.77

Table 1: Turnover in dollars over time for Train and Test data

Strategy	Train data	Test data
Autoregression mean reversion	3423733.999	1040806.634
Simple moving average crossover	58268.058	28115.981
Moving average mean reversion	920745.911	141915.956

Table 2: Turnover in number of units over time for Train and Test data

Plots of ΔV_t, ΔV_t^{cap}, and ΔV_t^{total} and plot their accumulated values (i.e., cumsum()) on Train and Test set:

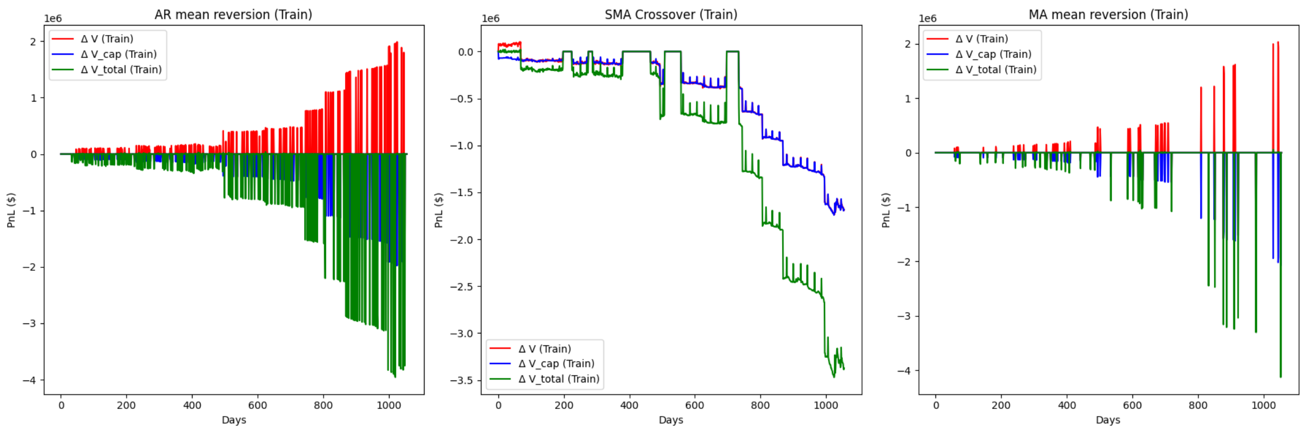


Figure 9: Pnl Series on train data

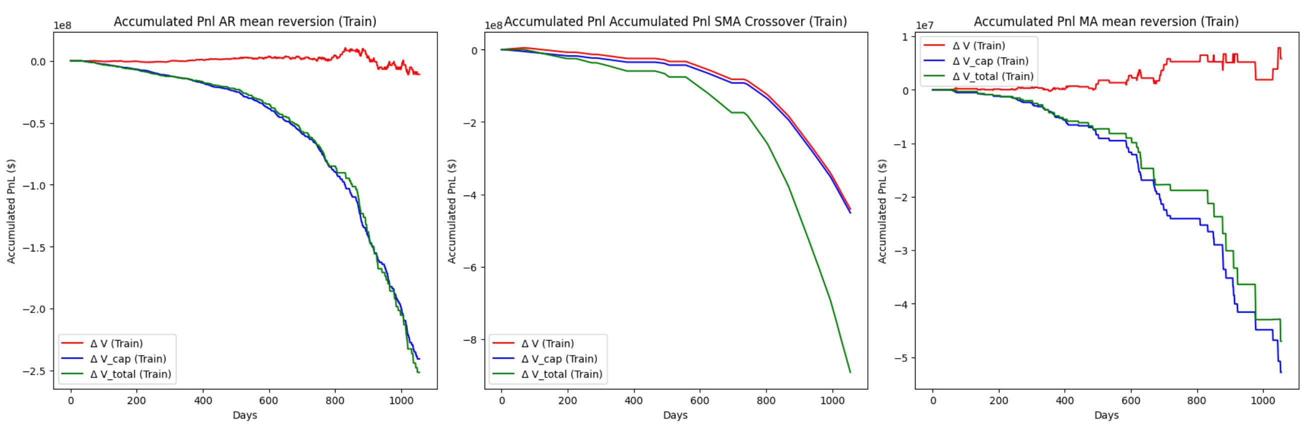


Figure 10: Accumulated Pnl Series on train data

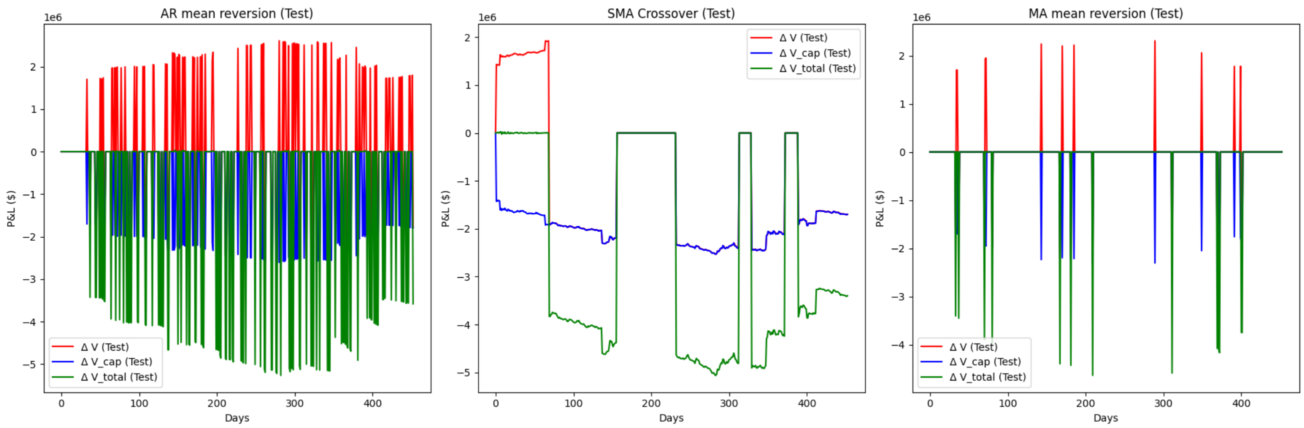


Figure 11: Pnl Series on test data

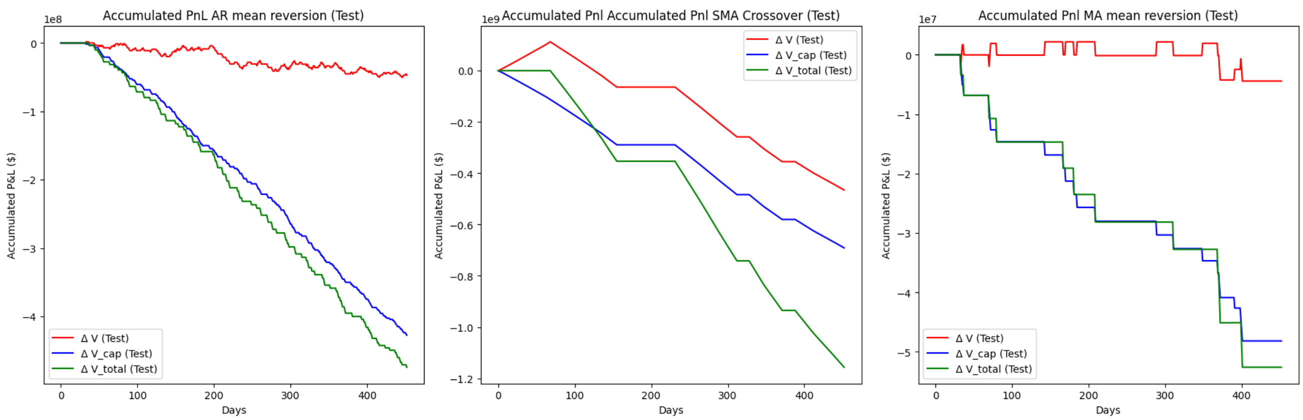


Figure 12: Accumulated Pnl Series on test data

Performance indicators

Table 3 and Table 4 show the results of the three strategies on the performance indicators - Sharpe Ratio, Sortino Ratio, Maximum drawdown, and Calmar Ratio.

Strategy	Sharpe Ratio	Sortino Ratio	Max drawdown	Calmar Ratio
AR mean reversion	0.43397	0.88206	-0.19365	31.51812
SMA crossover	0.03322	0.03768	-3.19198	146.1260
MA mean reversion	0.17739	0.17833	-0.17568	31.54756

Table 3: Performance metrics for the three strategies using Train data

Strategy	Sharpe Ratio	Sortino Ratio	Max drawdown	Calmar Ratio
AR mean reversion	0.46318	1.36174	-0.25393	32.62858
SMA crossover	0.03405	0.03829	-1.33447	53.26424
MA mean reversion	0.14718	0.32272	-0.07350	53.87255

Table 4: Performance metrics for the three strategies using Test data

Average excess returns of the strategies in the test set versus their standard deviations, and the SPDR average excess return and its standard deviation.

The following figure shows the plot of average and standard deviation daily excess returns PnL of the three strategies on the Test set. It also shows the average and standard deviation of excess returns of SPDR which is the `excess_return` column in the Test set.

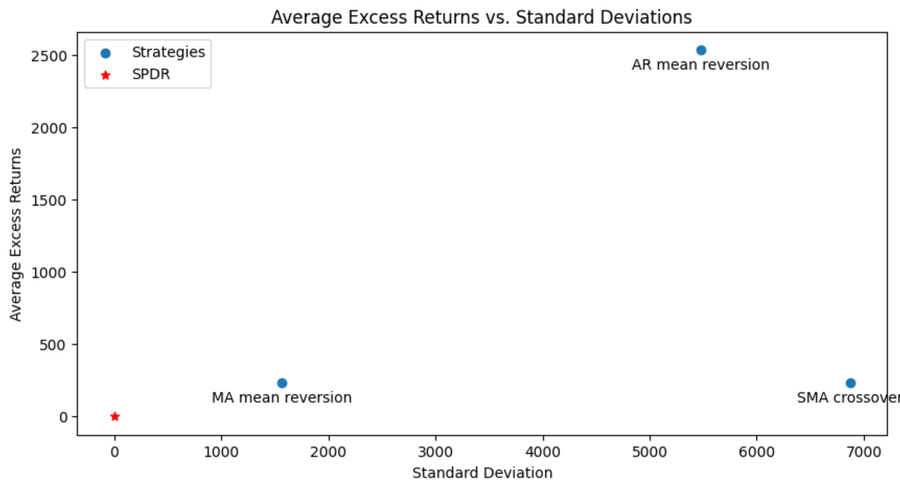


Figure 13: Average excess returns vs Standard deviations

Drawdown chart over time for all of the strategies separately

$$DD_t = \max[PnL_s] - PnL_t$$

and the historic rolling 90-day volatility of the underlying asset (p_t) on each chart.

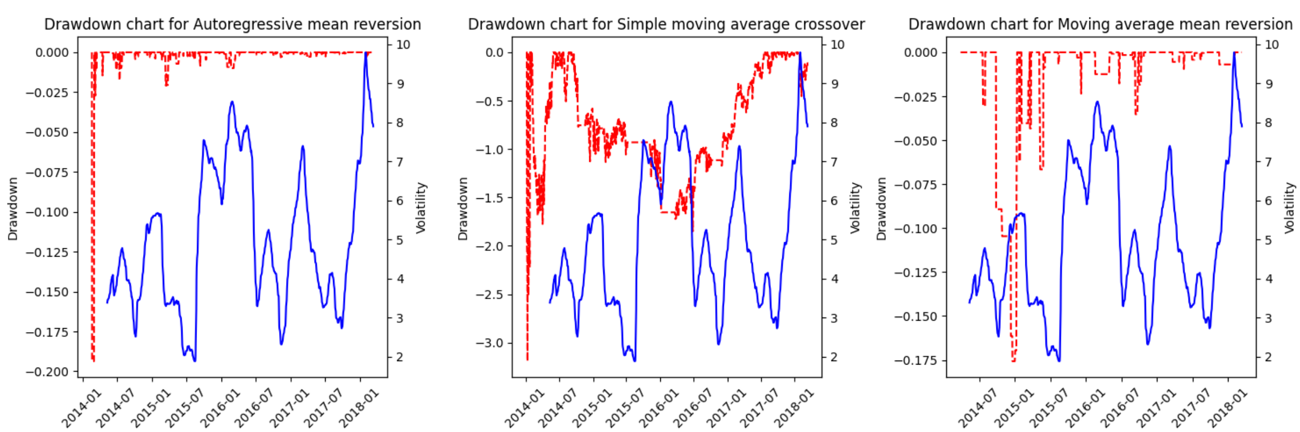


Figure 14: Drawdown chart of the three strategies on Train data

0.4 Discussion

From the PnL series plots, we can infer that only two strategies have accumulated positive values of PnL V_t on the train data which are Autoregressive mean reversion and Moving average mean reversion. The AR mean reversion also gives positive returns on the test data, however, the accumulated total V_t^{total} and accumulated capital $V_t^{capital}$ go down with time which suggests that they generate an overall loss like the rest. This suggests that none of the strategies could have been profitable on this data and need further research and tweaking with the hyperparameters.

A higher Sharpe ratio, Sortino Ratio and Calmar Ratio indicate a better risk-adjusted return. However, a lower maximum drawdown indicates a better strategy.

Overall from the performance indicators, the Autoregression mean reversion strategy appears to be the best performer among the three strategies, both for the train data and the test data. It has the highest Sharpe Ratio, Sortino Ratio, and Calmar Ratio, and the lowest maximum drawdown, -0.19365 and -0.25393, for both train and test data respectively. On the other hand, the Simple moving average crossover strategy appears to be the worst performer, with the lowest Sharpe Ratio, Sortino Ratio, and Calmar Ratio, and the highest maximum drawdown for both train and test data. The MA mean reversion strategy appears to perform better than the Simple moving average crossover strategy but not as well as the AR mean reversion strategy.

Analysis of Drawdown Charts:

1. The biggest drawdowns occur when the drawdown line (represented by the red dashed line in Figure 16) drops to its lowest point, indicating the largest percentage loss from the previous peak. This is the period when the strategy underperforms due to the volatile behavior of the stock market data. It occurs mostly around 2015 for SMA crossover and MA mean reversion strategies.

2. This can be related to the volatility as it can be seen in Figure 14 that the time series data experienced high volatility in the 2015-2016 period of time for the SMA crossover strategy and the MA mean reversion strategy. This can be explained by the fact that increased market volatility might result in larger price fluctuations and potentially more significant losses for the trading strategy.

3. The use of margin can boost a trading strategy's potential profits while simultaneously raising its risk of losses. Therefore, less margin may be used in safer techniques, depending on the underlying market volatility. For example, if the market is highly volatile, it may be riskier to use high levels of margin, as this could magnify losses in a drawdown. On the other hand, if the market is relatively stable, using higher levels of margin may be less risky, as the potential losses may be lower. Mean-reversion strategies like autoregressive mean reversion and moving average mean reversion may require less margin than trend-following strategies like simple moving average crossover tend to have lower drawdowns and less volatility. However, the success of each strategy is greatly influenced by the state of the market, thus margin usage must be regularly assessed and modified as necessary.

Bibliography

[1] S. Kapil and J. Gupta. Performance characteristics of hedge fund indices. *Theoretical Economics Letters*, 9:2176–2197, 2019.

[2] The moving average crossover strategy: A study. *Inspira- Journal of Modern Management & Entrepreneurship (JMME)*, 10(03):141–146, 2020.

[3] Peng Huang and Tianxiang Wang. On the profitability of optimal mean reversion trading strategies. 2016.