

Volatility Voyage: Assignment 2

Part I :

In the previous assignment, you have learnt the logic of Simple Moving Average (SMA) and Exponential Moving Average (EMA) strategies. These trend-following methods rely purely on price dynamics without consideration for the market's volatility. Moving ahead, we take a step further and see how integrating volatility into our trading strategy behaves.

In this question, you will be needed to

1. Implement three sequentially evolved trading strategies in python code (logic described below) and
 2. Build a backtesting framework (as discussed in the meet) and evaluate the strategies coded.
-

Strategy Design

Strategy 1: Basic EMA Crossover Strategy

Your job in the first part is to implement the EMA crossover strategy. First generate a 14-period Exponential Moving Average (EMA) of the closing price. We use this to identify potential entry and exit points.

- **Buy Signal**
 - If the current closing price is greater than the 14-period EMA, and the closing price on the previous period was less than the EMA, then it's a buy signal.
- **Sell Signal**
 - If the current closing price is less than the 14-period EMA, and the closing price on the previous period was greater than the EMA, then it's a sell signal.
- **Zero Signal**
 - If the condition hasn't changed from the previous day (no crossover), append a 0, do not signal any action.

Strategy 2: EMA Strategy with Dynamic ATR Band

Let's now build on Strategy 1 by aiming to capture stronger market signals. When the price movement, (the difference between the current and previous close, or more generally the difference between the current closing price and the 14-period EMA) is greater than the current volatility, it indicates a higher probability of a strong signal to take a trade. To better identify these strong market trends, we'll introduce two bands, each representing:

- **Buy Band** = $\text{EMA} + (\text{Current High} - \text{Current Low})$
- **Sell Band** = $\text{EMA} - (\text{Current High} - \text{Current Low})$

Signal Generation :

- **Buy Signal**
 - If the price crosses above the upper band, indicates a buy signal.
- **Sell Signal**
 - If the price crosses below the lower band, it indicates a sell signal.
- **Zero Signal**
 - If the price stays within the band, hold the position.

Strategy 3: ATR Breakout Strategy

We can improvise strategy 2 by incorporating market volatility as measured by the Average True Range (ATR) into the strategy and taking trades with help of it. Here, we compare the price to a dynamically adjusted band based on the ATR.

- First, compute **True Range (TR)** for each day. TR is the maximum of:
 - Current High - Current Low
 - Absolute value of (Current High - Previous Close)
 - Absolute value of (Current Low - Previous Close)
- Then take a 14-day EMA of the TR values to get the ATR.

The core idea here is that if a **price move exceeds recent average volatility**, a potential trend may be forming indicating an insightful signal.

- **Buy Band** = $\text{EMA} + k \times \text{ATR}$
- **Sell Band** = $\text{EMA} - k \times \text{ATR}$

Usually $k = 1.5$ (can be tuned for obtaining better results)

Signal Generation :

- **Buy Signal**
 - If current closing is higher than upper band, indicating a breakout to the upside.
- **Sell Signal**
 - If current closing is lower than the lower band, it's a downside breakout.
- **Zero Signal**
 - If neither condition is met, hold the current position.

Backtesting Framework

Code a backtesting engine which is able to calculate the above backtesting parameters given in the signals column in the dataframe and outputs the following dataframes.

Refer : [Strategy+SignalGen](#) [BacktestingSample.ipynb](#)

Trade Wise DataFrame:

The trade wise data frame is a trade log and should be containing the following columns for each trade:

- Entry Index
- Exit Index
- Entry Date
- Exit Date
- Trade Duration
- Returns for the trade in percent
- Type of Trade: long or short
- Max Drawdown for the Trade
- Max Dip for the Trade
- Quantity or number of stock traded

Daily DataFrame:

The daily dataframe would be containing a daily log of the following parameters.

- Portfolio Value: Also plot this
- Number of Stock or Quantity held
- Profit from initial capital till now in percent

Add Graphs to gain valuable insights and mark the entry and exit points.

Metrics :

Using the data frames generated, compute the following metrics:

- Benchmark Returns (%)
- Net Profit (%)
- Gross Profit (%)
- Number of Trades (Total, Wins, Losses)
- Maximum and Average Holding Time
- Maximum and Average Drawdown (%)
- Maximum and Average Dip(%)
- Sharpe Ratio (assume risk-free rate = 0)

Part II :

Now, we move on to incorporating some more financial metrics to better understand our portfolio's [risk averseness](#).

The two new metrics to be incorporated in your results is Value at Risk(VaR) and Conditional Value at Risk(CVaR).

VaR: VaR is a risk measure that estimates the maximum loss you might expect on an investment over a certain time period (like one day or one month), at a given confidence level. It tells you the threshold loss that will not be exceeded with a certain probability.

VaR is defined with respect to a certain confidence level. For example, a 95% confidence level means we are 95% sure that the loss will not be worse than the VaR number. In other words, on 95 days out of 100, the loss should be less than or equal to the VaR.

Think of it in terms of a certain **percentile** of your losses arranged in order.

CVaR: also called Expected Shortfall, measures the average loss you can expect on the worst days when losses exceed the VaR threshold. It helps understand how bad losses could be beyond the VaR limit.

If the CVaR at 95% confidence is ₹15,000, it means that on those worst 5 days (when losses are bigger than ₹10,000), the average loss is ₹15,000.

For the given task, to calculate VaR and CVaR work with the **Daily Dataframe**.

Hedging and portfolio sizing: In this part, you will extend your volatility trading strategies by incorporating a dynamic portfolio sizing approach based on the India VIX index.

Task:

1. **Load the India VIX data - Dataset.**
2. **Calculate the mean and standard deviation** of the India VIX over the available period.
3. **Define two VIX thresholds** based on the mean and standard deviation:
 - Threshold 1 = Mean + 0.5 × Std Dev
 - Threshold 2 = Mean + 1.5 × Std Dev
4. **Implement a dynamic capital allocation scheme** for your trading strategy based on daily India VIX values:
 - When India VIX < Threshold 1, allocate 100% of capital to trades.
 - When Threshold 1 ≤ India VIX < Threshold 2, allocate 75% of capital to trades.
 - When India VIX ≥ Threshold 2, allocate 50% of capital to trades.
5. **Modify your backtesting code** to reflect the changing capital allocation and adjust the number of shares or contracts traded accordingly each day.
6. **Evaluate the performance** of your strategy with dynamic portfolio sizing and compare it with the fixed 100% capital allocation from Part I.

Provide key metrics such as net returns, drawdowns, and Sharpe ratio, VaR,CVaR

To be noted :

For both questions, work with two stocks Adani Green, Ticker:ADANIGREEN.NS and Bharat Electronics Limited. Ticker: BEL.NS

Date: 22/05-2024-22/05/2025