Quant Assignment 2

December 15, 2023

Important Instructions

- It is preferable that you code the indicators allotted to you in the first question and try to use the same indicators for the second question also, but if someone finds their allotted Technical Indicators hard to code, then he/she can try to code other indicators, but in that case, we expect the person to do it with perfection. Students submitting their own allotted Technical indicators will be given extra brownie points.

- Submission deadline is **17/12/2023 11:59 p.m**.

- Late submitters will be awarded a penalty this time, which will be taken into consideration while finalizing the **ratification** list.

- All of you must submit your `.ipynb` file after running all the cells.

- You have to attempt all the questions.

Q1)

You have to submit a `.ipynb` file containing the Python code for the technical indicators (1 from each indicator basket) and a document containing the explanation for the chosen parameters (for example, lookback time period, smoothing period, open/high/low/close, etc.). Obviously, since you have to tell the parameters means that you have to check out the results by playing with the default parameters of the indicators to identify which parameters give out the best results. TradingView is one of the best platforms to see more about various technical indicators and how they change their behavior on changing their parameters.

Logistics: It's an individual exercise, if your roll number ends with 0 / 1, code the first indicators of each basket, if your roll number ends with 2 / 3, code the second indicators of each basket, and so on. Take the dataset which was provided in the form of a **CSV** file in the previous assignment.

Momentum Indicators

1. Average Directional Index

2. MACD

3. Stochastic Oscillator

4. RoC

5. Ichimoku cloud lines

Volume Indicators

1. Volume Price Trend Indicator

2. Volume RSI

3. Accumulation and Distribution Line Indicator

4. Negative volume index

5. Ease of movement indicator

Volatility Indicators

1. Donchian Channels

2. Average True Range

3. Implied Volatility

4. Parabolic SAR

5. Keltner Channels

Q2)

Design and Implement a Trading Strategy using Technical Indicators in Python

Background: Financial markets are dynamic and influenced by various factors. Traders often rely on technical analysis to make informed decisions. Technical indicators, such as moving averages, Relative Strength Index (RSI), and Bollinger Bands, provide valuable insights into market trends and potential trading opportunities.

Task in hand: The goal of this question is to develop a Python-based trading strategy that leverages popular technical indicators to generate buy or sell signals in financial markets. The strategy should be able to analyze price data given in the **CSV** file that was provided in the last assignment, apply the indicators that were allotted to you in **Q1**, and then execute trading decisions based on predefined conditions.

Key Components:

1. **Indicator Calculation:** Incorporate the calculation of the technical indicators that are allotted to you.

2. **Signal Generation:** Develop the logic for generating buy or sell signals based on the computed technical indicators. Define clear and robust rules for triggering trading decisions.

3. **Risk Management:** Implement risk management strategies to control the size of trades and manage potential losses. Consider incorporating features such as stop-loss orders.

4. **Backtesting:** Create a comprehensive backtesting framework to evaluate the performance of the trading strategy using the CSV data. Assess the strategy's profitability, risk-adjusted returns, and other relevant metrics.

5. **Optimization:** Explore opportunities for optimizing the strategy parameters to enhance its overall performance. This may involve parameter tuning or the introduction of additional indicators.

6. **Documentation:** Provide clear documentation for the implemented trading strategy, including the rationale behind signal generation rules, and any considerations related to risk management (e.g., your rationale behind the metrics you chose for your stop loss).

Deliverables: You should provide a `.ipynb` file containing your trading strategy and it should also include buy/sell signal columns for the data. You should also provide and present various key metrics such as Sharpe Ratio, Sortino Ratio, No. of Winning and Losing Trades, Maximum Drawdown, Total Profit, Net Profit.

Q3)

Implementing Single Candlestick Pattern Recognition in Python

Background: Candlestick patterns are essential tools in technical analysis for understanding market sentiment. A single candlestick can provide valuable information about the potential direction of price movement. Recognizing and interpreting these patterns can be crucial for making informed trading decisions.

Task in hand: The goal of this question is to identify "**The Hammer Formation**'. For more information regarding this candlestick pattern, you can refer to The Technical Analysis chapter of the Zerodha Varsity resource which can also be found in the following link: https://zerodha.com/varsity/chapter/single-candlestick-patterns-part-3/. You can also refer to other sites such as Investopedia for all of your doubts not necessarily related to this question only.

Deliverables: You should provide a `.ipynb` file containing your code for identifying the pattern depending upon your own decided metrics about the '**Shadow to Real body**' ratio you seem fit. You should calculate and then show the number of times the pattern is identified and you should show at least one candlestick which is a Hammer by plotting it with the help of many Python Libraries such as MatPlotLib, Plotly etc depending upon your own choice and comfort.