Quantitative Trading Strategy: BTC/USDT Market

Y Harsha Vardhan
B.Tech, IIT Bombay
Roll Number: 24b1069

July 28, 2025

Contents

1. Introduction

The cryptocurrency market, particularly Bitcoin (BTC), has witnessed significant volatility and growth in recent years. This dynamic environment offers ample opportunities for algorithmic trading strategies that can systematically identify and exploit market inefficiencies.

The objective of this project is to design, implement, and evaluate an algorithmic trading strategy for the BTC/USDT (Bitcoin to Tether) trading pair using historical daily price data. The goal is to generate profitable trading signals—`LONG`, `SHORT`, or `EXIT`—based on quantitative analysis, while effectively managing risk and ensuring robustness across different market regimes.

This project explores classical and modern technical indicators for feature engineering, applies regime-based strategy logic, and backtests the performance of the trading algorithm using realistic constraints such as transaction fees and position management. Performance is assessed using key metrics like total return, Sharpe ratio, maximum drawdown, and the total number of trades.

2. Dataset Description

The dataset used in this project consists of historical daily candlestick data for the BTC/USDT trading pair. Each record represents one day and contains the following columns:

- **datetime** — The date of the trading day.
- **open** — Price of BTC at the start of the day.
- **high** — Highest price of BTC during the day.
- **low** — Lowest price of BTC during the day.
- **close** — Price of BTC at the end of the day.
- **volume** — Trading volume during the day.

The data spans from 2019 to early 2023 and provides a broad view of both bullish and bearish market conditions, making it suitable for training and testing robust trading strategies. The dataset was synthetically generated for academic purposes, simulating realistic price behavior while avoiding any real-world biases or proprietary data concerns.

Basic preprocessing included parsing datetime fields, checking for missing or duplicated records, and computing derived metrics such as daily returns and rolling averages. The dataset serves as the foundation for all signal generation and backtesting in the project.

3. Feature Engineering

We engineered the following key feature:

- **Average True Range (ATR): Measures market volatility over a 14-day window.**

Additionally we calculate:

- **Volume Spike: A dynamic threshold based on the mean and standard deviation of the past 5-day volume.**

These features are used to trigger entries, exits, and trailing stop thresholds within the strategy.

4. Strategy Design

The strategy design is rule-based and comprises the following logic:

1. **Entry Conditions:**
   - **A trade is considered only if there is a significant volume spike.**
   - **If the candle is bullish (close > open) → enter long.**
   - **"If bearish (close < open) → enter short."**
2. **Trailing Stop:**
   - **A dynamic stop-loss is set at 2xATR below (for long) or above (for short) the entry price.**
3. **Exit Conditions:**
   - **Exit or reverse if price moves against the position for 3 consecutive days.**
   - **Exit if price hits the dynamic trailing stop.**
   - **Reverse if a volume spike occurs with a candle in the opposite direction.**

4.1. Trading Regimes

The system supports three primary trading regimes based on the signal values:

1. **Neutral (0)**: No trade is executed; the system holds cash
2. **Directional Entry (+1/-1)**: A new trade is initiated when the system is not in a position
3. **Directional Exit and Reversal (+2/-2)**: The current position is closed and immediately reversed.

This structure allows both trend-following and mean-reversion strategies to be implemented depending on the signal generation logic.

4.2. Signal Generation Logic

The signals are generated externally and provided via a CSV input. The signal column contains discrete integer values:

1. **1**: Open a long position (buy)
2. **-1**: Open a short position (sell)
3. **2**: Reverse to long
4. **-2**: Reverse to short
5. **0**: Hold / Do nothing

The system supports modular signal input, which allows easy integration with any custom indicator or machine learning strategy. The assumption is that signals are based on historical price/volume data and optionally enhanced with indicators like RSI, MACD, or moving averages.

4.3. Risk Management Rules

Risk is managed using:

- **Take-Profit (TP)** and **Stop-Loss (SL** thresholds, specified per trade row.
- **Transaction fees** of 0.15% per trade to account for slippage and exchange costs.
- **Signal Validation**: If an invalid signal is received (e.g., attempting to open a long position while already long), the system throws an exception, ensuring consistency.

These mechanisms help prevent runaway losses and simulate real-world conditions.

4.4. Position Sizing

Position sizing is based on a fixed trade capital amount (default $1000), with optional compounding enabled. In compounding mode, profits or losses from previous trades are added to the capital base, affecting subsequent trade size.
The quantity is not in terms of units/shares but in dollar exposure (positive for long, negative for short), which abstracts the model for any asset class.

5. Backtesting Framework

The backtesting engine simulates trades using minute-level data. Core assumptions include:

- **Latency-free execution** at closing price or specified TP/SL level.
- **Sequential evaluation**: Signals and price data are matched minute-by-minute.
- **Realistic fees** of 0.15% per trade are deducted
- **TP/SL monitoring** happens at every minute granularity using a "master file" that provides high/low prices for accurate simulation.

The engine captures realistic market behavior without relying on future information, avoiding look-ahead bias. It supports both static and compound returns, drawdown tracking, and cumulative PnL calculations.

6. Results and Evaluation

6.1. Performance Metrics

The system calculates:

- Net Profit
- Gross Profit and Loss
- Win Rate
- Average Win/Loss
- Maximum and Average Holding Time

- Total Trades
- Long vs Short Distribution

These statistics offer a holistic view of the strategy's performance.

6.2. Profit and Sharpe Ratio

Profitability is reported in both absolute and percentage terms over the backtest duration. The Sharpe Ratio is calculated assuming a risk-free rate of 0, scaled to a 365-day basis.
This measures risk-adjusted performance.

6.3. Drawdown Analysis

Drawdowns are computed using capital over time. The maximum and average drawdowns reflect the largest equity declines during the trading period, offering insights into capital risk.
Visualizations show drawdown periods in red for better interpretability.

6.4. Trade Frequency and Distribution

The trade frequency depends on the signal generation logic. Distribution across long and short trades is tracked, and winning vs losing streaks are counted to assess consistency.
Trade durations are also evaluated to understand how long positions are typically held.

7. Discussion

The backtesting framework successfully models realistic trading conditions and accurately reflects the strategy's strengths and weaknesses. Challenges encountered included:

- Correctly modeling TP/SL triggers minute-by-minute
- Ensuring trade direction consistency
- Handling edge cases with back-to-back conflicting signals

One limitation is the assumption of instant execution with no slippage beyond transaction fees, which may not hold in illiquid markets.

8. Conclusion

This system provides a robust, modular, and transparent framework for testing trading strategies. With flexible input signals and TP/SL controls, it supports a wide variety of regime-based models. Results show that proper signal tuning, risk control, and sizing dramatically influence performance and drawdown.

9. Strategy's Performance Indicators

The following are the strategic performance indicators for my Strategy in the main.py file.

- `Total Trades:` 104
- `No. of Long Trades:` 51
- `No. of Short Trades:` 53
- `Benchmark Return(%):` 325.632937277405
- `Benchmark Return (on $1000):` 3256.32937277405
- `Win Rate:` 42.30769230769231
- `Sharpe Ratio:` 2.2613287387209655

10. Future Work

Potential improvements include:

- Introducing leverage and margin requirements.
- Supporting partial position exits or scaling in/out.
- Adding slippage models or market impact simulation.
- Automating walk-forward optimization and hyperparameter tuning.
- Real-time deployment on paper trading platforms like Binance Testnet.

References

- Pandas TA Documentation
- TA-Lib Technical Indicators
- **Backtesting strategies and evaluation metrics from quant research blogs and online sources**
  - QuantStart: Successful Backtesting of Algorithmic Trading Strategies
  - QuantStart: Event-Driven Backtesting with Python (Part I)
  - QuantInsti: Performance Metrics, Risk Metrics, and Strategy Optimization