

- Introduction
- Research Question
- Data
  - Data Description
  - Key Metrics
    - Summary Statistics
- Timeline
  - Data Preprocessing
  - Model: Fair value gap (FVG)
    - FVG Detection Logic
    - Position Sizing
    - Parameters
  - Model: Machine Learning and Neural Networks
- Evaluation
  - Performance Metrics
    - Sharpe Ratio
    - Maximum Drawdown
  - Benchmark Comparison
    - Outperformance vs. Buy-and-Hold
  - Trade Analysis
  - Tentative Targets

# Introduction

---

The cryptocurrency market, particularly Bitcoin, presents unique challenges for forecasting due to its high volatility and difficulty to predict. This project explores an algorithmic trading approach to formulate a trading strategy for bitcoin. I aim to develop a framework that can effectively predict optimal trading actions (buy, sell, or hold) and their respective magnitudes (for buy and sell) based on historical market data.

# Research Question

---

How can we develop a robust daily decision-making framework for Bitcoin trading that effectively determines optimal actions (buy, sell, or hold) and their magnitude to maximize returns while managing risk across different market conditions?

Starting with an initial budget  $B_0$ , an investor makes a sequence of daily trading decisions over a period of  $T$  days, denoted as  $D = \{d_1, d_2, \dots, d_T\}$ . For each day  $t$ , the decision  $d_t \in \mathbf{R}$  represents the amount of Bitcoin to trade, where:

- $d_t > 0$  indicates buying  $d_t$  units of Bitcoin
- $d_t < 0$  indicates selling  $|d_t|$  units of Bitcoin
- $d_t = 0$  indicates holding the current position

Given the daily Bitcoin prices  $P = \{p_1, p_2, \dots, p_T\}$ , the investor's portfolio value  $V_t$  after day  $t$  can be expressed as:

$$V_t = B_t + Q_t \cdot p_t$$

Where  $B_t$  is the remaining cash budget and  $Q_t$  is the quantity of Bitcoin held after day  $t$ . These values are updated according to each decision:

$$\begin{aligned} B_t &= B_{t-1} - d_t \cdot p_t - c(d_t, p_t) \\ Q_t &= Q_{t-1} + d_t \end{aligned}$$

With  $c(d_t, p_t)$  representing transaction costs associated with the trade.

The objective is to find the optimal decision sequence  $D^*$  that maximizes the final Sharpe Ratio while maintaining acceptable drawdown levels. Here, each decision where each decision implicitly relies on forecasting future Bitcoin price movements based on available market information up to day  $t$ . Note, that we can use the closing price on day  $t$  as the buying/ selling price because crypto markets operate 24/7.

At any given time there may be excess cash that the algorithm has decided not to invest into bitcoin. Return on this excess cash will be based on that day's risk free rate, which would provide a lower bound on average return from alternate investments that are independent of bitcoin.

## Data

---

## Data Description

---

The analysis will utilize a dataset of daily bitcoin data from 2015 to present, sourced through the [cctx](#) library.

# Key Metrics

---

## Summary Statistics

	btc_close	btc_volume
mean	22819.45	14398.3
std	24479.6	11759.82
min	211.16	683.8
25%	4004.59	6739.0
50%	10789.38	11355.56
75%	37266.84	18243.23
max	106159.26	165542.81

## Timeline

---

## Data Preprocessing

---

- Exploratory Data Analysis (EDA)
- Time series decomposition to understand trends, seasonality, and noise
- Feature engineering to create technical indicators
- Correlation analysis to identify relationships between Bitcoin and other assets

## Model: Fair value gap (FVG)

---

A Fair Value Gap (FVG) occurs when there is a significant gap between price candles that indicates a potential area where price may return to "fill" the gap. The strategy identifies and trades based on these market inefficiencies.

## FVG Detection Logic

The strategy identifies FVGs using the following criteria:

1. Analyzes three consecutive candles (A, B, C)
2. For Bullish FVG:
  - Candle C's high is below Candle A's low
  - Gap must be larger than average candle body size × multiplier
3. For Bearish FVG:
  - Candle C's low is above Candle A's high
  - Gap must be larger than average candle body size × multiplier

## Position Sizing

Position sizes are dynamically calculated based on multiple factors:

1. **Base Position:** Starts with 30% allocation
2. **Volatility Adjustment:** Up to 30% additional based on 20-day rolling volatility
3. **Momentum Component:** Up to 20% based on 20-day price momentum
4. **Gap Size Impact:** Up to 20% based on FVG size
5. **Consecutive Signals:** Position size increases with consecutive signals (20% per signal, max 100%)

The final position size is capped at 100% of available capital and is further adjusted by:

- Upside scaler for buy positions
- Downside scaler for sell positions

## Parameters

- **lookback\_period**: 20 days (for volatility calculation)
- **body\_multiplier**: 1.5 (minimum gap size relative to average candle body)
- **back\_candles**: 50 (historical candles to analyze)
- **test\_candles**: 10 (forward-looking period for signal validation)

# Model: Machine Learning and Neural Networks

---

## Evaluation

---

The project will evaluate the effectiveness of our trading strategy's decisions using the following performance metrics that directly assess how well our algorithm determines the optimal quantities ( $d_i$  values) to buy, sell, or hold:

## Performance Metrics

---

### Sharpe Ratio

Measures the excess return per unit of risk generated by our trading decisions. It is given by the formula:

$$sharpe = \frac{R_p - R_f}{\sigma_p}$$

where:

- $R_p$  is the annualized portfolio return resulting from our sequence of trading decisions ( $d_i$ )
- $R_f$  is the risk-free rate (3-month Treasury yield)
- $\sigma_p$  is the annualized standard deviation of our portfolio's periodic returns

### Calculation process:

1. For each time period  $t$ , the algorithm generates a decision  $d_t$  (quantity to buy/sell)
2. This updates our Bitcoin holdings  $Q_t$  and cash balance  $B_t$
3. Portfolio value at each step:  $V_t = B_t + Q_t \times p_t$
4. Calculate periodic returns:  $r_t = \frac{V_t - V_{t-1}}{V_{t-1}}$
5.  $R_p = (1 + \frac{\sum_{t=1}^n r_t}{n})^{365} - 1$
6.  $\sigma_p = \sqrt{\frac{\sum_{t=1}^n (r_t - \bar{r})^2}{n-1}} \cdot \sqrt{365}$

# Maximum Drawdown

Measures the largest peak-to-trough decline in portfolio value resulting from our trading decisions. It is given by the formula:

$$MaxDD = \frac{\max(\max(V_s) - V_t)}{\max(V_s)} \quad \forall s \leq t$$

Where  $V_t$  is the portfolio value at time  $t$ , directly resulting from our sequence of decisions.

## Calculation process:

1. For each time period  $t$ , our algorithm generates a decision  $d_t$  (quantity to buy/sell)
2. This updates our Bitcoin holdings ( $Q_t$ ) and cash balance ( $B_t$ )
3. Calculate portfolio value at each step:  $V_t = B_t + Q_t \cdot p_t$
4. Track the running maximum portfolio value:  $Peak_t = \max(V_1, V_2, \dots, V_t)$
5. Calculate drawdown at each step:  $DD_t = (Peak_t - V_t)/Peak_t$
6. Maximum drawdown is the largest drawdown observed:  $MaxDD = \max(DD_1, DD_2, \dots, DD_T)$

A lower maximum drawdown indicates our algorithm is generating  $d_t$  values that effectively reduce exposure before significant price drops

This directly evaluates if our model correctly identifies high-risk periods and adjusts position sizes accordingly

## Benchmark Comparison

---

### Outperformance vs. Buy-and-Hold

Compares the performance metrics (sharpe ratio and maximum drawdown) of our dynamic trading strategy against a simple buy-and-hold approach

- Buy-and-hold would be represented by  $D_h = \{\frac{B_0}{p_1}, 0, 0, \dots, 0\}$
- Our strategy generates a different sequence of  $d_t$  values that varies exposure over time
- The resulting Sharpe ratio and Drawdowns will be different and this comparison will be the final evaluation

# Trade Analysis

---

Examine individual trading decisions to evaluate their effectiveness through the following metrics:

- Win rate: Percentage of  $d_t$  decisions that contributed positively to portfolio value
- Average profit/loss per trade: Mean portfolio value change resulting from each non-zero  $d_i$
- Average holding period: Typical duration between buy and sell decisions

This analysis helps identify whether our algorithm makes consistently good quantity decisions or relies on a few outsized winners.

These metrics will be mainly used on the test set to understand what the model is really doing so I might find ways of improving it or making necessary changes.

## Tentative Targets

---

Prior to model development I am setting the following targets (which are subject to adjustment) for evaluating whether the approach was successful or not:

1. Sharpe ratio at least 30% higher than a buy-and-hold strategy over the test period.
2. Maintains a maximum drawdown at least 20% lower than a buy-and-hold strategy.
3. Win rate of at least 55%