End-Term Report: Algorithmic Trading

BTCUSDT and ETHUSDT Analysis

Team ID: 32

December 5, 2024

Contents

1 Executive Summary

This report presents the development and implementation of algorithmic trading strategies for BTC/USDT and ETH/USDT cryptocurrency markets. Leveraging a Linear Regression model alongwith Decision Tree, our approach achieved significant returns compared to benchmark strategy while maintaining robust risk management measures. The report details the end-to-end process, including data acquisition, preprocessing, strategy design, backtesting, and optimization.

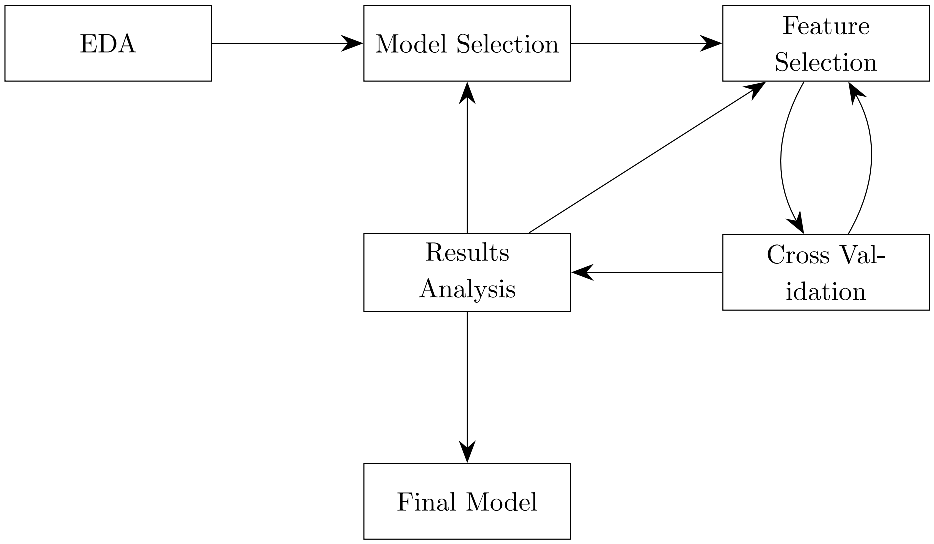

Figure 1: Our Modelling approach

2 Introduction

The primary objective was to create statistically sound trading strategies for BTC/USDT and ETH/USDT that deliver superior returns while effectively managing risks. The development process encompassed:

1. **Data Preprocessing:** We denoised the data using HA candles. We also added many features based on volume, trend, momentum, volatility, price, range and oscillators.
2. **Model Selection:** We implemented Decision Tree Classifier and Linear Regression models to predict future returns and for different coins and timeframes, we selected the best model.
3. **Feature Selection:** We used correlation analysis, plotting and PCA to select the most relevant features for the model.
4. **Risk Management Techniques:** We implemented stop loss, HMM and drawdown analysis to manage risks.
5. **Backtesting:** We used the SDK to backtest our strategies on historical data to evaluate their performance.
6. **Results Analysis:** We analyzed the results of the backtesting to understand the performance of the model and the strategies by using various plots and metrics on different quarters of the data.

7. **Hyperparameter Tuning using Cross Validation:** We used k-fold cross validation to tune the hyperparameters of the model. The k-fold cross validation was used to prevent overfitting and better generalization of the model on different datasets.

3 Challenges and Solutions

In our development process, we encountered following primary challenges:

1. **Feature Selection:** The most challenging aspect was identifying an optimal set of features for our model which could generalise on different timeframes and coins. We observed that different set of features were performing well on different coins and timeframes. We tried to address this challenge by making using different feature selection techniques like correlation analysis, plotting, PCA etc.

2. **Adaptation of Model to different Market Conditions:** A big challenge for us was to make the model adapt to different market conditions that is bullish and bearish markets. We tried to address this challenge by using regime analysis.

4 Strategy Overview

4.1 For BTC:

4.1.1 Strategy for Long Periods(1 day)

Initially, similar to ETH, we experimented with linear regression; however, it did not yield promising returns. Subsequently, we employed Decision Trees to generate positions (long, neutral, short), which were then utilized to create trading signals. The strategy balance generated from these signals closely matched the benchmark returns.

To enhance performance, we incorporated the Decision Tree's output as an additional feature in the linear regression model. This combination significantly reduced drawdowns, increased the Sharpe ratio, and improved overall strategy returns. Furthermore, by integrating Heikin-Ashi (HA) candles into the analysis, the strategy successfully outperformed the benchmark returns.

Statistical tests were conducted to evaluate the significance of the features, ensuring robustness in feature selection. The final features used in the linear regression model were as follows:

- Logarithm of HA close values
- Logarithm of HA open values
- Logarithm of HA high values
- Logarithm of HA low values
- Suggested positions derived from the Decision Tree

4.1.2 Strategy for Short Periods(1 hour)

Ensemble of signals
Sometimes we had multiple signals which we used to merge into one final signal,

- **Precedence vote of signals:** This is the case that we have multiple signals, we check that if one of the signal is dominant (or strong) in one direction then one can reliably use that signal, this precedence is ensembeled.

- **Average vote of signals:** For the cases that there is no clear preference of which signal to prefer, we use average of signals to make the final signal.

4.1.3 Linear Regression Model

The strategy predicts short-term returns and generates actionable signals:

- **Prediction Mechanism:** Model predicts short-term returns using past data.
- **Threshold-based Action:** Signals are generated if predictions cross predefined thresholds.

4.1.4 Key Features for regression

1. **Volume and Dollar Volume:** Reflective of market liquidity.
2. **Z-Score of Price:** Measures deviation from mean price levels.
3. **Lagged Returns:** Incorporate temporal dependencies.
4. **Squared Returns:** Highlight periods of extreme volatility.

4.2 For ETH:

4.2.1 Core Strategy

The trading strategy implemented for this coin is grounded in linear regression and HMM(Hidden Markov Model), with the primary objective of predicting future returns of ETH/USDT by using uncorrelated features selected using feature selection algorithms.

4.2.2 Linear Regression Model with HMM

The strategy predicts short-term returns and generates actionable signals :

- **Prediction Mechanism:** The model predicts short-term returns using past data encoded in the form of features. It leverages historical trends and patterns to enhance the accuracy of return forecasts.
- **Volatility-Prediction:** HMM is used to predict how to place trades based on the volatility of the market. By analyzing hidden states, it helps identify patterns in market behavior, enabling informed and strategic trading decisions..
- **Threshold-based Action:** Threshold-based action combines linear regression predictions and volatility assessments from an HMM. A signal is triggered only when the linear regression output surpasses set thresholds, and the HMM predicts volatility to be within acceptable limits, ensuring safer decision-making.

4.2.3 Insights

- **Threshold Signals:** Predictions exceeding a positive threshold trigger buy signals, while negative values below a threshold trigger sell signals.
- **Market Conditions in 2023:** The year 2023 was marked by low market activity and reduced volatility, making it unsuitable for effective trading strategies based on our model analysis.

4.2.4 Key Features for Regression

1. **Z-Score of Price**
2. **Open and Close**
3. **Return**
4. **Z-Score**
5. **High-Low**
6. **RSI**
7. **Chikou Span**
8. **Squared Returns**

5 Indicators Explored

5.1 Volume Indicators

Volume indicators reflect the trading activity and liquidity in the market. They help gauge the strength of a trend or detect potential reversals.

- **Extraordinary Volume (`extraordinary_volume_eth/btc`):** Flags periods where trading volume significantly exceeds the rolling mean volume by more than three standard deviations. High extraordinary volume indicates potential market-moving activity.
- **Liquidity (`liquidity_eth/btc`):** The product of trading volume and closing price. It reflects market depth and the ability to absorb large trades.
- **Log Liquidity (`log_liquid_eth/btc`):** The logarithm of liquidity to stabilize its scale for modeling purposes.
- **Volume Weighted Average Price (VWAP) (`VWAP_eth/btc`):** Represents the average price weighted by trading volume, commonly used as a benchmark to evaluate trade executions.
- **Money Flow Multiplier (MFM) and Volume (MFV) (`MFM_eth/btc`, `MFV_eth/btc`):** Calculate money flow, considering the relationship between price movements and volume.
- **Chaikin Money Flow (CMF) (`CMF_eth/btc`):** Combines MFM and cumulative volume over a period (e.g., 50) to measure buying or selling pressure.

5.2 Trend Indicators

Trend indicators help identify the direction and strength of market trends over various timeframes.

- **Mean Price (`mean_price_eth/btc`):** The rolling average price over a window (e.g., 20 periods), providing a smoothed trendline.
- **Price Deviation (`price_deviation_eth/btc`):** The difference between the current price and its rolling mean. It signals overbought or oversold conditions.

- **Exponential Moving Averages (EMA) (`ema_eth/btc`):** Computes a weighted average of past prices with exponentially decreasing weights, emphasizing recent price data.
- **Chikou Span (`chikou_span_eth/btc`):** A lagging line from the Ichimoku Cloud system, representing the current closing price shifted back 26 periods to gauge momentum.
- **Directional Movement Index and Average Directional Index (ADX) (`adx_eth/btc`):** The ADX measures trend strength based on the difference between positive and negative directional movement.

5.3 Momentum Indicators

Momentum indicators assess the speed and magnitude of price changes, helping to identify overbought or oversold conditions.

- **Relative Strength Index (RSI) (`RSI_eth/btc`):** Measures the magnitude of recent price changes to identify overbought (>70) or oversold (<30) conditions.
- **Stochastic RSI (`Fast_K_eth/btc`, `Fast_D_eth`, `Slow_K_eth/btc`, `Slow_D_eth/btc`):** Combines RSI with stochastic calculations to measure RSI's position relative to its high/low range.
- **Momentum (`MOM_eth/btc`):** The difference between the current price and the price 26 periods ago, indicating the strength of a trend.
- **Moving Average Convergence Divergence (MACD) (`MACD_eth/btc`, `MACD_Signal_eth/btc`, `MACD_Hist_eth/btc`):**
  - *MACD Line:* Difference between the 12-period and 26-period EMAs.
  - *Signal Line:* A 9-period EMA of the MACD line.
  - *MACD Histogram:* The difference between the MACD line and the signal line.
- **Directional Momentum Multipliers (`multiplier_eth/btc`):** Categorizes ATR-based momentum into levels for better trading signal generation.

5.4 Volatility Indicators

Volatility indicators measure the rate of price changes, which can signal potential breakouts or reversals.

- **True Range and Average True Range (ATR) (`true_range_eth/btc`, `ATR_eth/btc`, `ATR_Ratio_eth/btc`):**
  - *True Range:* The maximum of (high-low, high-previous close, low-previous close).
  - *ATR:* The moving average of the true range over a specific period (e.g., 14).
  - *ATR Ratio:* ATR divided by the rolling average of ATR over a longer period.
- **Rolling Volatility (`Rolling_Volatility_eth/btc`):** Standard deviation of returns over a specific window.
- **Z-Score (`z_score_eth/btc`):** Measures the number of standard deviations the current price is from the mean.
- **Bollinger Bands (`BB_upper_eth/btc`, `BB_lower_eth/btc`, `BB_middle_eth/btc`, `BB_bandwidth_eth/btc`, `BB_percent_eth/btc`):**

- Upper, lower, and middle bands are calculated based on a moving average and standard deviation.
- Bandwidth measures band width relative to the moving average.
- Band percentage measures the price's position within the bands.

5.5 Oscillators

Oscillators measure market momentum and potential reversal points by oscillating within a bounded range.

- **Williams %R (`williams_eth/btc`):** A momentum indicator that measures overbought or oversold levels relative to the high-low range over 14 periods.
- **Stochastic Oscillator Crossover (`Fast_Crossover_eth/btc`, `Slow_Crossover_eth/btc`):** Detects when the %K line crosses the %D line, signaling potential buy/sell opportunities.

5.6 Price and Range Indicators

These indicators analyze price relationships and range patterns to predict breakouts or consolidations.

- **High-Low Range (`high_low_eth/btc`, `high_low_square_eth/btc`):** Measures the absolute and squared difference between high and low prices.
- **True Range Components (`tr_high_low_eth/btc`, `tr_high_close_eth/btc`, `tr_low_close_eth/btc`):** Break down components of the true range calculation.

5.7 Composite Indicators

Composite indicators combine multiple features to provide a more holistic view of the market.

- **Hidden State Models (`hidden_state`):** Derived from Gaussian Hidden Markov Models, they classify market states as high or low volatility.
- **Fibonacci Band Levels (`band_btc`):** Derived from high and low prices, these levels indicate potential support and resistance zones.

6 Correlation Analysis

Below are the clusters of the features we obtained during correlation analysis:

Cluster	**Features**
0	`extraordinary_volume_eth`
1	`mean_price_eth`, `ATR_eth`, `average_ATR_last_100_eth`, `BB_upper_eth`, `BB_lower_eth`, `BB_middle_eth`, `chikou_span_eth`, `Lowest_Low_eth`, `Highest_High_eth`, `ema_eth`, `VWAP_eth`
2	`price_deviation_eth`, `z_score_eth`, `CMF_eth`
3	`returns_square_eth`, `liquidity_eth`, `log_liquid_eth`, `high_low_eth`, `high_low_square_eth`, `tr_high_low_eth`, `tr_high_close_eth`, `tr_low_close_eth`, `true_range_eth`
4	`ATR_Ratio_eth`
5	`multiplier_eth`, `position_signal_eth`

Cluster	Features
6	RSI_eth, MACD_Hist_eth, BB_percent_eth, Fast_%K_eth, Fast_%D_eth, Slow_%K_eth, Slow_%D_eth
7	MACD_eth, MACD_Signal_eth, MOM_eth
8	BB_bandwidth_eth
9	Fast_Crossover_eth
10	Slow_Crossover_eth
11	Fast_Normalized_Distance_eth, Slow_Normalized_Distance_eth
12	adx_eth

7 Threshold analysis

Threshold is the value for which a linear regression signal strength of more than threshold is considered a signal to take position.

7.1 Threshold as a function of other features

The idea is that threshold should be different across time as the linear regression's output will be higher in cases when the return is higher.

1. **Variable threshold :** We used threshold of the form $a\times$ volatility $+b$
2. **Variable threshold :** We also used threshold of the form $a\times$ average returns $+b$

7.2 Hyperparameter Tuning

We used rolling cross validation (10 validations) to choose the correct hyperparameters across both the coins and in conclusion this turned out to be better.

8 Ensemble of signals

Sometimes we had multiple signals which we used to merge into one final signal, this is because some strategies outperformed in some timeframes while the other dominated in other timeframes, to combine these we used the following features :

8.1 Decision trees

Decision trees are basically if-else branches which are trained to identify which signals work well in which conditions. This is also one of the main models that we used in low frequency.

8.2 Precedence vote of signals

This is the case that we have multiple signals, we check that if one of the signal is dominant (or strong) in one direction then one can reliably use that signal, this precedence is ensembeled.

8.3 Average vote of signals

For the cases that there is no clear preference of which signal to prefer, we use average of signals to make the final signal.

9 Performance Metrics

9.1 Historical Performance Analysis

Year	Benchmark (%)	Our Returns (%)
2021–22	390.574246	1437.276891
2022–23	-67.878809	-80.036041
2023–24	90.249632	578.453184

Table 2: Performance Comparison of Benchmark and Our Returns for ETH

Year	Benchmark (%)	Our Returns (%)
2021–22	56.920444	-15.449288
2022–23	-64.638147	162.548343
2023–24	155.551527	829.323824

Table 3: Performance Comparison of Benchmark and Our Returns for BTC

9.2 Key Metrics

Metric	BTCUSDT	ETHUSDT
Trades Executed	339	292
Profit(%)	1581.180104	31214.222941
Sharpe Ratio	1.260373	1.793681
Maximum Drawdown	85.632381	83.733933
Benchmark Return(%)	544.324298	1578.237463
Time to Recovery	117.125000 days	49.541667 days

Table 4: Performance Metrics on Given Data (2020-23)

Metric	BTCUSDT	ETHUSDT
Trades Executed	18	18
Profit	1766.869234	875.594857
Sharpe Ratio	8.202687	11.70401
Maximum Drawdown	6.349987	25.770774
Benchmark Return(%)	155.551527	90.249632
Time to Recovery	50.041667 days	91.541667 days

Table 5: Performance Metrics on Test Data (2023)

10 Unique Approaches and Innovations

10.1 Decision Tree and Linear Regression Loop

For experimental purposes, a model was designed utilizing an iterative loop of Decision Trees and Linear Regression. Initially, the outputs of the Linear Regression model were set to zero. The Decision Tree model used the RSI of Heikin-Ashi (HA) close values and the outputs of the Linear Regression model to generate positions. These positions, along with the previous Linear Regression outputs, were then used to train a new Linear Regression model.

With each iteration, the Decision Tree and Linear Regression models refined one another, progressively improving the overall performance of the strategy and leading to enhanced returns.

Note: *Decision tree predicts position while linear regression predicts returns which are used to predict signals*

11 Learning Outcomes

The most significant learning point was the importance of **interpretability and adaptability** in algorithmic trading:

- **Statistical Edge:** Even marginal predictive power ($R^2 \approx 0.01$) can translate to profits when applied consistently.
- **Data Quality:** Noise significantly impacted model reliability, emphasizing the importance of preprocessing.
- **Feature selection** ML might tend not to generalize, maybe overfit for different features and parameters.

12 Conclusion and Recommendations

- It is important to keep track of progress and all the models. It is better to have a pipelined procedure from the start.

References

We have added our source code in the submission zip file

A Features Incorporated:

A.1 In BTC

```
features = [
    f"open_{coin}",
    f"returns_square_{coin}",
    f"return_{coin}",
    f"high_low_square_{coin}",
    "log_return"
]
```

Figure 2: Features for BTC

A.2 In ETH

```python
features = [
    f"open_{coin}",
    f"close_{coin}",
    f"returns_square_{coin}",
    f"return_{coin}",
    f"z_score_{coin}",
    f"high_low_{coin}",
    f"high_low_square_{coin}",
    f"RSI_{coin}",
    f"chikou_span_{coin}"
]
```

Figure 3: Features for ETH

B Performance Curves

Figure 4: Performance Curve for BTC

Figure 5: Performance Curve for ETH

C Additional Strategies

We also tried the following strategies while exploring different indicators and trading models.

C.1 Decision Tree

We discovered that incorporating non-linear features into Linear Regression did not yield favorable results. Therefore, we decided to leverage the capabilities of decision trees to better utilize these signals. Using this approach, we generated the optimal positions for the training period.

Pipeline Overview

1. **Data Preparation:**
   - We determined the optimal positions for the training period by utilizing the shifted returns and applying a threshold to decide whether to enter a position or remain neutral.
   - Hidden states are calculated, likely representing latent market conditions to inform trading decisions.
2. **Feature Engineering:**
   - A combination of technical indicators and predictions from a decision tree (`dec_tree_btc`) is used as inputs for further modeling.
   - Selected features are refined iteratively using linear regression results (`lr_res`).
3. **Decision Tree Model:**

- A `DecisionTreeClassifier` with a maximum depth of 5 is trained to classify positions (e.g., buy, sell, or hold signals).

4. **Linear Regression Model:**
   - Predicts shifted returns (`shifted_return_btc`) using a subset of engineered features.
   - Predictions (`dec_tree_btc`) from the decsion tree are also used for prediction

C.1.1 SDK Backtest Results

- **Sharpe Ratio:** 3.6628
- **Sortino Ratio:** 8.8260
- **Trades Executed:** 318
- **Leverage Applied:** 2.0
- **Profit Percentage:** 40,340.14%
- **Maximum PnL:** $217,572.08
- **Minimum PnL:** -$39,771.99
- **Final Balance:** $404,401.41
- **Total Fees:** $60,833.08

C.1.2 Observations and Conclusion:

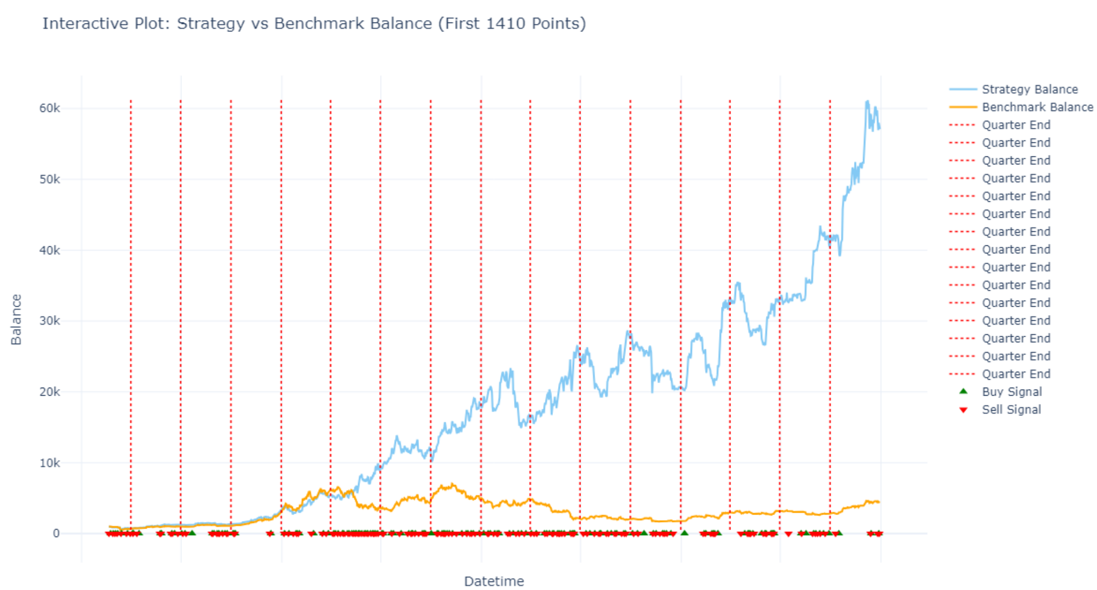

Figure 6: Performance Curve for BTC 1d

The strategy has exceptionally well over the years and has generated compounded revenues heavily surpassing the benchmark returns.