KHARAGPUR DATA SCIENCE HACKATHON

Algorithmic Trading Model Development for BTC/USDT Crypto Market

Team : 21je0001

Team Members

Aadya Dewangan
Arpita Kargaonkar
Aditya Mukherjee

Table of Contents

Introduction

Problem Description

The problem consists of developing algorithmic trading models for the BTC/USDT cryptocurrency market, aiming to outperform benchmark returns.

It is required to create trading algorithms that can generate returns while managing risk effectively in the specific BTC/USDT market.

Methodology

Data Preparation

The analysis was performed on BTC/USD OHLCV data (O: Open, H: High, C: Close, V: Volume) every four hours. We filter out rows with zero volume to avoid meaningless data points. Next, we check for any NaN values, and remove all such occurences. Also, we make a list of our time series data, namely, the open, high, low and close prices.

Trading Logic and Associated Functions

The goal is to optimize buying and selling times for Bitcoin (BTC) to maximize profit. Various strategies involve analyzing indicators like Exponential Moving Average (EMA), Relative Strength Index (RSI), and Bollinger Bands. Time series forecasting models, such as AutoRegressive Integrated Moving Average (ARIMA) and Long Short-Term Memory (LSTM), are also employed. However, it's crucial to recognize the inherent volatility of cryptocurrency markets, making adaptability, prudent risk management, and staying informed vital for successful trading.

Shorting Condition

Current High Price > High prices of n past candles and
Current High Price > High prices of n future candles

Buying Condition

Current Low Price<Low prices of n past candles
Current Low Price < Low prices of n future(predicted) candles

Motivation For Strategy

Executing trades based on identifying the highest high or lowest low within a specified candle range offers profit potential. This strategy involves going long or short, maintaining trades only when criteria are met, and avoiding stop-loss or take-profit. It incorporates efficient risk management while aiming for profit.
To address non-stationarity in predicting future prices, we used the Long Short Term Memory (LSTM) Model, using the past 25 time periods to predict future values.
Predictive accuracy was assessed using RMSE as well as R2 score, indicating a strong alignment between predicted and actual data. This blend of a the fields of machine and finance forms a robust framework, requiring ongoing vigilance in evolving market conditions.

Why not ARIMA?

1. Due to the highly fluctuating nature of the data the model was severely underfitting or overfitting
2. Hyperparameter tuning for the model did not help much

Why not Technical Analysis

1. Due to the highly complex nature of the data , there was a significant lag in the indicators and they were often not able to predict well
2. Backtesting Results did not so look promising either as they were coming out to be negative / in losses.

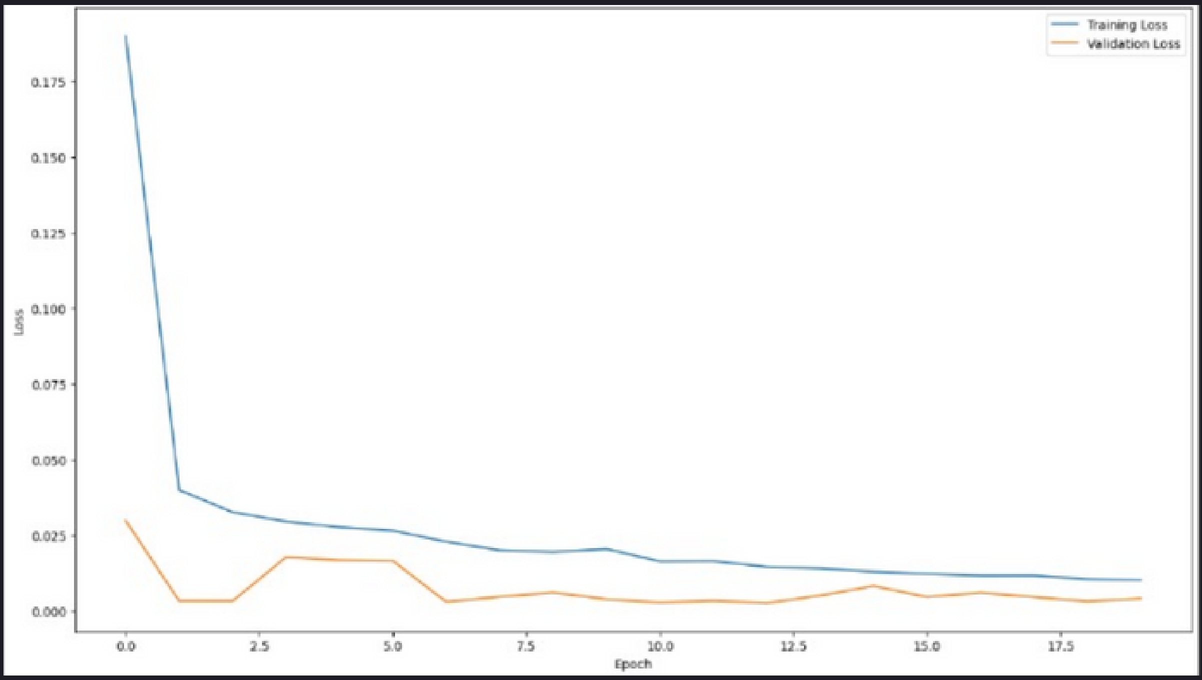
How is LSTM better?

1. They are better at capturing intricate patterns that may not meet the human eye
2. Great for predictions of data that has a relationship with respect to time

Optimizing the Model & Reducing Overfitting in LSTM Model

1. Due to the extreme fluctuations nature of the data, 2 models were created for each - high and low to increase the accuracy.
2. A uni-directional LSTM network was not able to capture the patterns, so a bidirectional or Bi - LSTM was used
3. The data values were scaled using Robust Scaler to prevent overfitting without losing the patterns in the data
4. Hyperparameter tuning for the LSTM Network

Model Training



TRAINING VS VALIDATION LOSS

LSTM PREDICTIONS VS ACTUAL DATA

Model Arcbitecture

layer type	units	activation
Bi-LSTM	128	relu
Dropout	0.3	
Bi-LSTM	64	relu
Dropout	0.3	
Bi-LSTM	32	relu
Dropout	0.3	
Bi-LSTM	16	relu
Dropout	0.3	
Bi-LSTM	1	linear

Model Prediction Accuracy

Overall R2 Score = 0.801

Backtesting of the strategy

Utilizing the Backtesting.py module, the evaluation of the trading strategy's performance hinges on two pivotal statistical metrics:

1. Sharpe Ratio:

- Definition: The Sharpe ratio is derived from the division of mean daily returns by the annualized volatility.
- Formula:
  Sharpe ratio = (mean daily returns)/(annualized volatility)
- Annualized Volatility: Calculated as the standard deviation of daily returns multiplied by the square root of 365 (number of trading days in a year).

2. Sortino Ratio:

- Definition: The Sortino ratio is determined by dividing the average excess return by the downside deviation.
- Formula:
  Sortino ratio = (Average excess return)/(Downside deviation)
- Average Excess Return: The average return above the risk-free rate over a specific period.
- Downside Deviation: The standard deviation calculated only for negative returns over the same period.

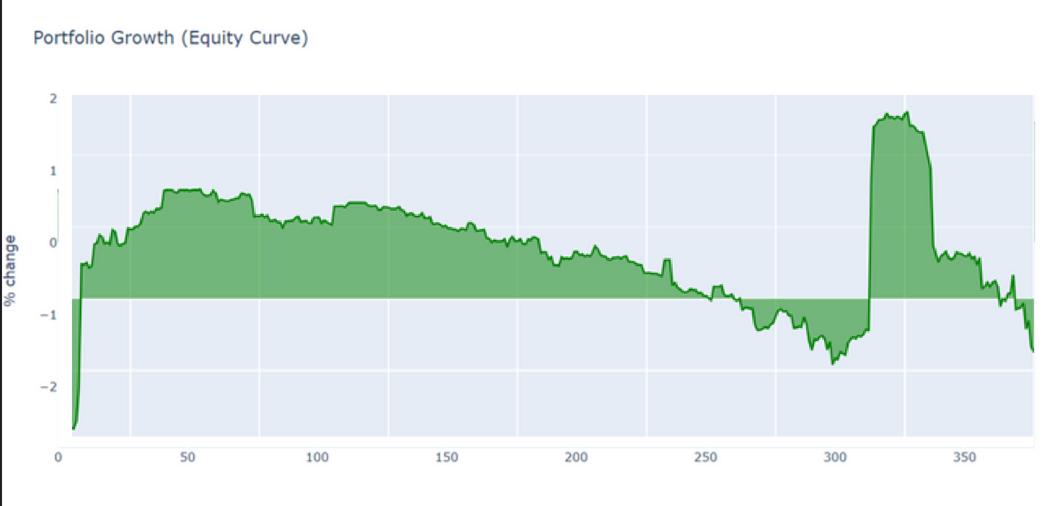
These ratios serve as critical benchmarks, gauging the risk-adjusted performance of the trading strategy. The Sharpe ratio accounts for the interplay between returns and volatility, while the Sortino ratio specifically hones in on downside risk by considering only negative returns. The assessment using these metrics offers valuable insights into the strategy's efficacy in generating returns relative to the associated risk.

Result

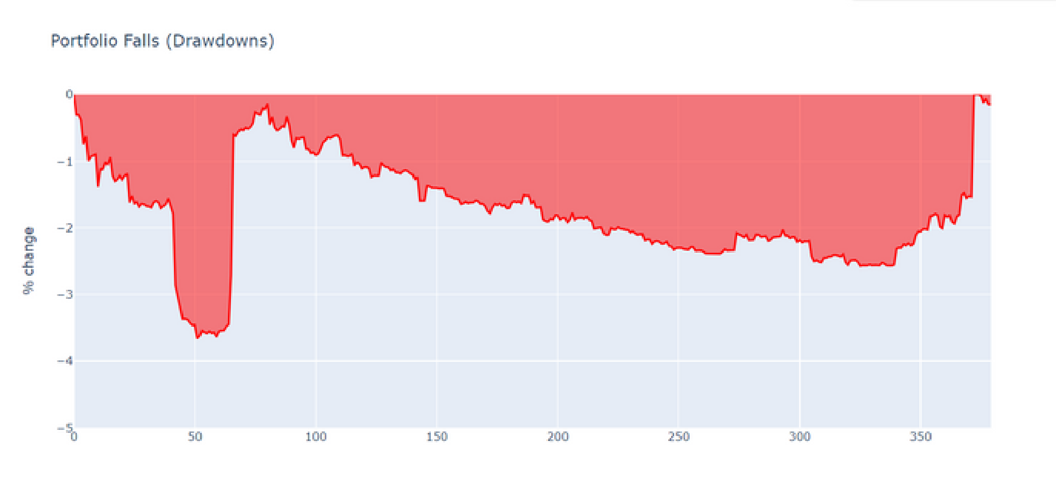
START	01-01-2018 05:30:00 AM
END	31-01-2022 05:30:00 AM
DURATION	1491 Days, 4h Data
INITIAL EQUITY	100,000
BUY & HOLD	97%
FINAL EQUITY	187,230
GROSS PROFIT	197094
NET PROFIT	154756
GROSS LOSS	42338
LARGEST WINNING TRADE	3190.30
LARGEST LOSING TRADE	2098.13
SHARPE RATIO	1.23
SORTINO RATIO	7.39
MAX DRAWDOWN	3.79%
AVG DRAWDOWN	2.58%
AVG WINNING TRADE	2034.49
AVG LOSING TRADE	1243.24
TOTAL CLOSED TRADES	2096

Backtesting Graphs

EQUITY CURVE



DRAWDOWNS



VOLATILITY