ZeltaLabs

Kharagpur Data Science Hackathon

Team- AmEx Aspirants

KHARAGPUR DATA ANALYTICS GROUP

KSHITIJ 19TH-21ST JAN 2024 TECHNO-MANAGEMENT FEST IIT KHARAGPUR

ZeltaLabs

OBJECTIVES:

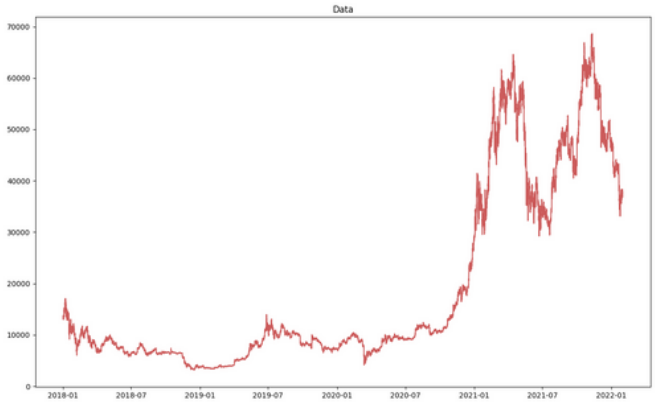
Developing Advanced Machine Learning Models for Algorithmic Trading in BTC/USDT

- **Forecasting Accuracy:** Create predictive models to accurately forecast BTC/USDT price movements.
- **Risk Management:** Implement strategies to maximize returns while minimizing risks.
- **Method Comparison:** Compare performance across various machine learning techniques - Time Series Analysis, Regression, Deep Learning, Evolutionary Algorithms.
- **Back-testing for validity:** Validate model effectiveness through rigorous back-testing against historical data.
- **Innovation Emphasis:** Utilize state-of-the-art techniques to drive innovation in predictive modeling.
- **Benchmark Outperformance:** Aim to surpass existing market benchmarks with advanced algorithmic strategies.

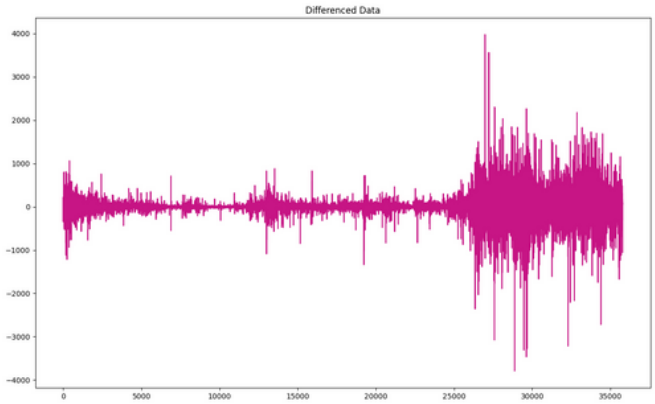
we want to leverage machine learning methods to establish new benchmarks in the dynamic and challenging crypto trading arena.

TIME SERIES ANALYSIS

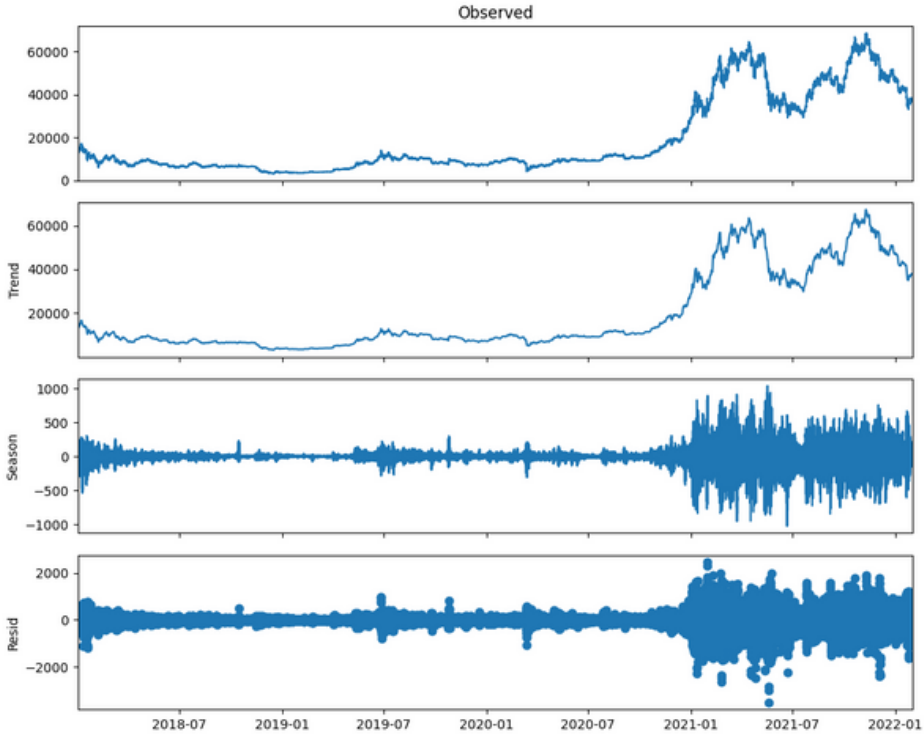
Time series analysis is a crucial part of any comprehensive analysis involving the stock market. It involves traditional, statistical methods to analyse a moving series and forecast what the future may hold.



The plots of the data and the differenced data are shown: the differencing was done to remove any trends, but what we end up getting is an almost completely white noise, as corroborated by the ADF test performed on both the data (with p-value statistic being 0.8 and 0 respectively).

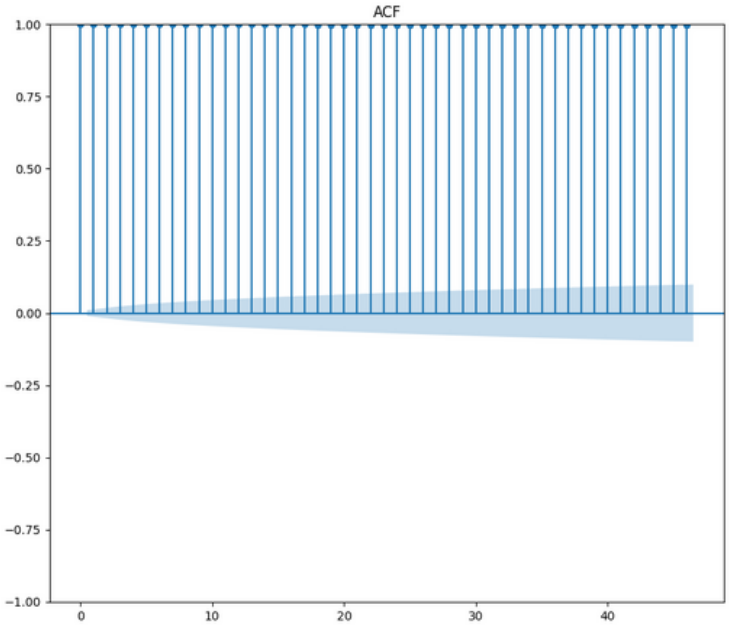


The stationarity condition is critical as a lot of processes assume stationarity as a prerequisite condition. The Augmented Dickey-Fuller test is a hypothesis test for the null hypothesis that a unit root exists (i.e. there is no stationarity).



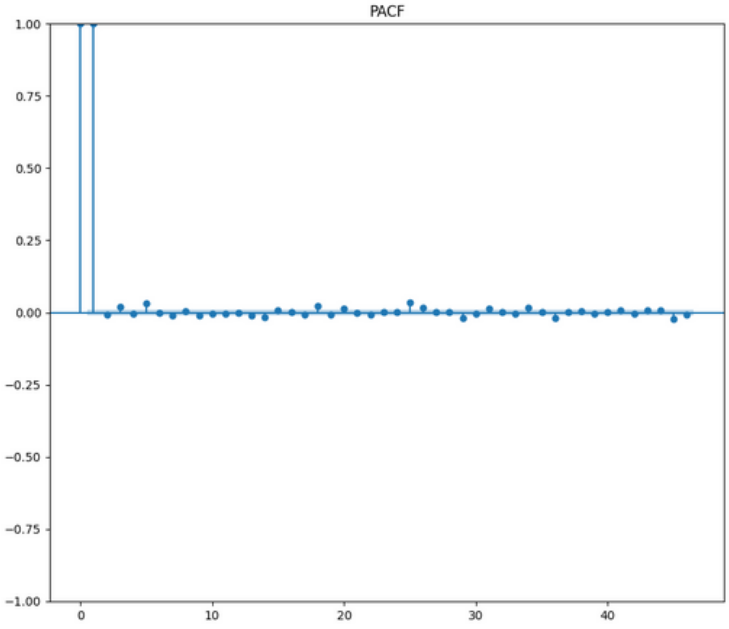
The seasonal decomposition paints an even grimmer picture: the STL decomposition (Seasonal-Trend decomposition using LOESS) shows the "trend" to be essentially the denoised data, and the seasonal and residual terms are all white noise so none of them exists at a significant level.

The seasonal decomposition process also revealed an inconsistency in the data, as we found out that a total of 121 hourly timestamps were missing in the btc_1h.csv file. Those rows were inserted and the values filled in by the method of linear interpolation.



The ACF and PACF graphs tell a lot: the ACF plot decays extremely slowly (going from 1.00 to 0.97 in 46 steps), indicating an extremely high trend.

On the other hand, the PACF plot cuts off after lag 2, indicating that some model containing the AR(2) might be useful in modeling the series.



A variety of ARIMA models were trained on the data, with **seasonality set to 0** (as the data is NOT seasonal). The goodness-of-fit was checked by the **Akaike Information Criterion,** which both rewards for accuracy and penalizes for unnecessary model complexity.

The minimum AIC was that of the ARIMA(2,0,2) model, which expectedly, has an AR order of 2.

The auto_arima() function present in the statsmodels module was also used, and the algorithm voted for ARIMA(3,0,3), which is not too unreasonable, given that the AIC of the model is not that far from the minimum, and the model performs better than ARIMA(2,0,2) which is expected, given its increased complexity.

Akaike Information Criterions:

MA/AR	0	1	2	3
0	431661.57732445985	431660.4062712174	431640.3991706522	431642.13748988707
1	431660.56141670246	431642.7672508065	431628.2499700096	431630.16534426267
2	431642.39743211446	431627.9817726457	431622.09805351094	431623.185325693

RANDOM FOREST REGRESSOR:

The Random Forest Regressor (RFR) is a sophisticated ensemble learning algorithm widely used in regression tasks. This method falls under the umbrella of decision tree-based algorithms and is a variant of the Random Forest algorithm, predominantly applied in classification scenarios. The core principle of RFR involves constructing multiple decision trees during the training phase. Each tree in the forest is built from a random subset of the data, ensuring diversity among the trees. This randomness not only helps in reducing overfitting but also enhances the model's generalization capabilities.

During the prediction phase, RFR aggregates the outputs of individual trees to arrive at a final prediction. This aggregation, often a mean or average of the predictions from all trees, contributes to a more accurate and stable result than what a single decision tree could provide. The ensemble approach effectively captures complex relationships in the data, making RFR particularly useful in scenarios with large, diverse datasets.

RANDOM FOREST REGRESSOR:

RFR's robustness to noise and its ability to handle non-linear relationships between variables are key strengths. Moreover, it requires relatively less tuning of parameters compared to other algorithms, making it user-friendly for practitioners. RFR is widely used in various domains, including finance for stock price prediction, in healthcare for disease progression modeling, and in environmental sciences for predicting weather patterns and climate change impacts.

In practice, the performance of RFR is often evaluated using metrics such as the R2 score, which reflects the proportion of variance in the dependent variable that is predictable from the independent variables. For instance, in a particular analysis with an 80:20 data split for training and testing, the RFR achieved an impressive R2 score of **0.9734674817368255**, indicating a high level of predictive accuracy on the dataset."

XGBOOST REGRESSOR:

XGBoost (Extreme Gradient Boosting) represents a powerful and efficient implementation of gradient boosting algorithms, a group of machine learning techniques known for their predictive accuracy and effectiveness. Originally conceptualized for tackling classification problems, XGBoost has been adeptly adapted to handle regression tasks through its XGBoost Regressor implementation.

At its core, XGBoost employs a gradient boosting framework. This approach involves sequentially adding predictors (decision trees), with each one correcting its predecessor. Unlike traditional boosting, where trees are added to correct the errors of previous trees, XGBoost optimizes this process using a gradient descent algorithm to minimize loss when adding new models. This method allows for the systematic handling of various types of regression problems, making it a versatile tool in the machine learning toolkit.

XGBOOST REGRESSOR:

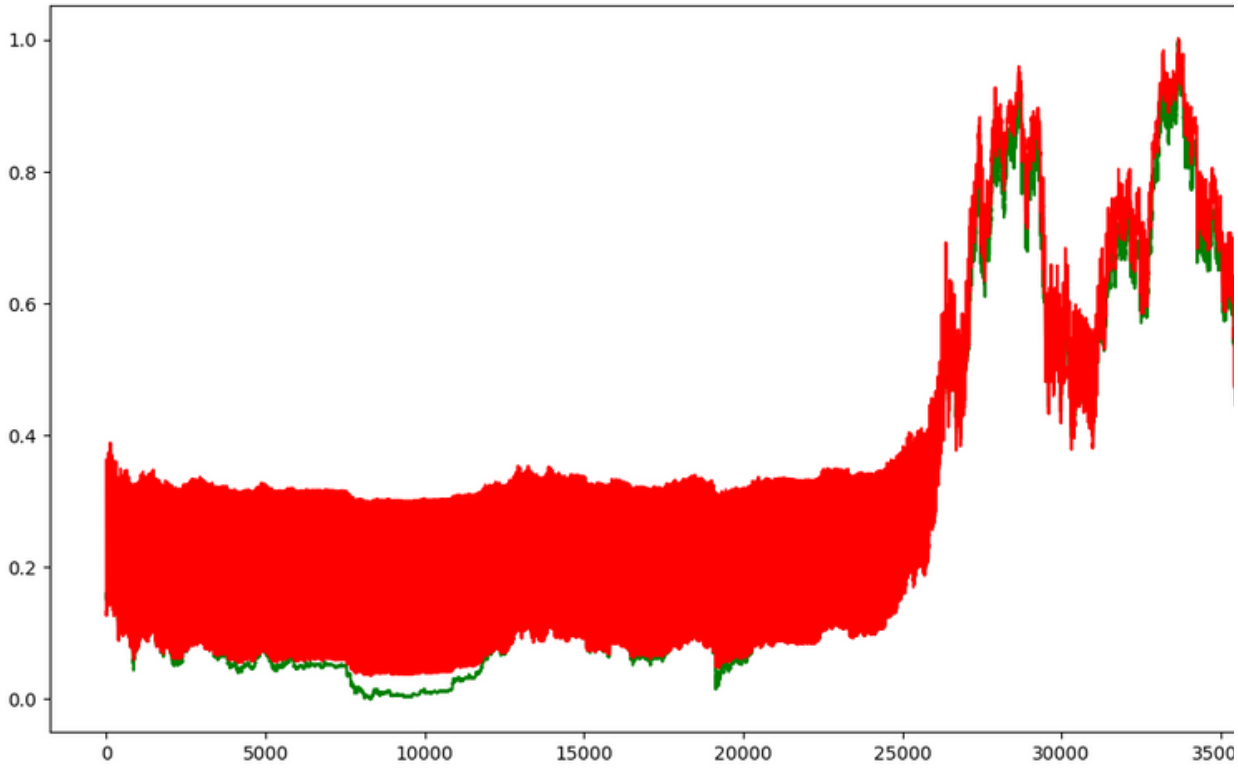
One of the key strengths of XGBoost is its scalability, which makes it highly efficient on large datasets. The algorithm has been designed to optimize computational resources, reducing the time required for training and prediction. XGBoost also includes built-in regularization features, which help prevent overfitting—a common challenge in machine learning models. Additionally, it supports various objective functions and evaluation criteria, making it adaptable to different regression scenarios.

Another notable aspect of XGBoost is its capacity to handle missing data and work with various types of data inputs. This flexibility, combined with its high performance, has made XGBoost a popular choice across diverse fields. It is widely used in finance for risk modeling, in e-commerce for demand forecasting, and in the field of medicine for predictive diagnostics.

In practical applications, XGBoost's effectiveness is often quantified using metrics such as the R2 score, which measures the proportion of variance in the dependent variable that is predictable from the independent variables. For instance, in a specific case where the algorithm was applied to a dataset with an 80:20 training-to-testing data split, the XGBoost Regressor achieved a R2 score of 0.9137534427536891. This high score indicates a strong predictive ability and accuracy in modeling the relationship between the independent and dependent variables in the dataset.

LONG SHORT-TERM MEMORY:

"LSTM (Long Short-Term Memory) networks represent a significant advancement in the field of recurrent neural networks (RNNs), particularly tailored for addressing sequence prediction problems. Traditional RNNs often struggle with learning long-range dependencies within data due to issues like vanishing or exploding gradients. LSTM networks, however, overcome these challenges with their unique internal structure, enabling them to retain information over extended periods.



The defining feature of LSTM networks is their intricate system of gates, including the input, forget, and output gates. These gates collaboratively manage the flow of information, determining what data should be retained or discarded as the network processes sequences. This functionality allows LSTMs to maintain a balance between remembering past information and adapting to new data, making them exceptionally suited for tasks that require the understanding of long-term dependencies.

LONG SHORT-TERM MEMORY:

Applications of LSTM networks are extensive and diverse, particularly excelling in areas such as language modeling, where they can predict the next word in a sentence; speech recognition, where they process spoken language for conversion to text; and time series prediction, where they analyze time-dependent data for forecasting future events.

In the context of stock price forecasting, the ResNLS architecture, as outlined in the paper 'ResNLS: An Improved Model for Stock Price Forecasting', integrates the strengths of LSTM with additional enhancements. This architecture likely incorporates elements designed to tackle the specific challenges of financial time series data, which are often non-linear, non-stationary, and highly volatile. The ResNLS model, by leveraging the memory and learning capabilities of LSTM networks, could potentially offer improved accuracy in predicting stock prices by efficiently capturing the temporal dependencies and patterns within market data.

The use of LSTM networks, particularly in sophisticated models like ResNLS, demonstrates the continuous evolution and applicability of neural networks in financial analytics. By incorporating LSTM's ability to remember and utilize historical information, the ResNLS model represents a promising approach in the realm of quantitative finance, offering potentially more accurate and reliable predictions in stock price forecasting."

MOVING AVERAGE CONVERGENCE DIVERGENCE(MACD):

MACD (Moving Average Convergence Divergence) is a crucial tool in the arsenal of technical traders and analysts, offering insights into market trends and momentum. This indicator is particularly valued for its dual functionality: trend following and momentum indication. The MACD is computed using moving averages, which are foundational tools in technical analysis, providing a smoothed representation of market trends.

The MACD indicator consists primarily of two components: the MACD line and the signal line. The MACD line is calculated as the difference between two moving averages of a security's price, typically the 12-period (fast) and 26-period (slow) exponential moving averages (EMAs). This line captures the essence of market momentum and trend direction. The signal line, on the other hand, is the EMA of the MACD line itself, usually over a 9-period span. It acts as a trigger for buy and sell signals.

MOVING AVERAGE CONVERGENCE DIVERGENCE(MACD):

Traders pay close attention to the interaction between these two lines, particularly the crossovers. A crossover occurs when the MACD line crosses above or below the signal line. A bullish crossover (MACD line moves above the signal line) suggests upward momentum, indicating a potential buying opportunity. Conversely, a bearish crossover (MACD line moves below the signal line) signals downward momentum, hinting at a possible selling point.

Another critical aspect of MACD analysis is the concept of divergence. Divergence occurs when the MACD line and the price chart of the security move in opposite directions. A bullish divergence, where the price hits lower lows while the MACD hits higher lows, can indicate a potential reversal of a downtrend. Similarly, a bearish divergence might signal the weakening of an uptrend.

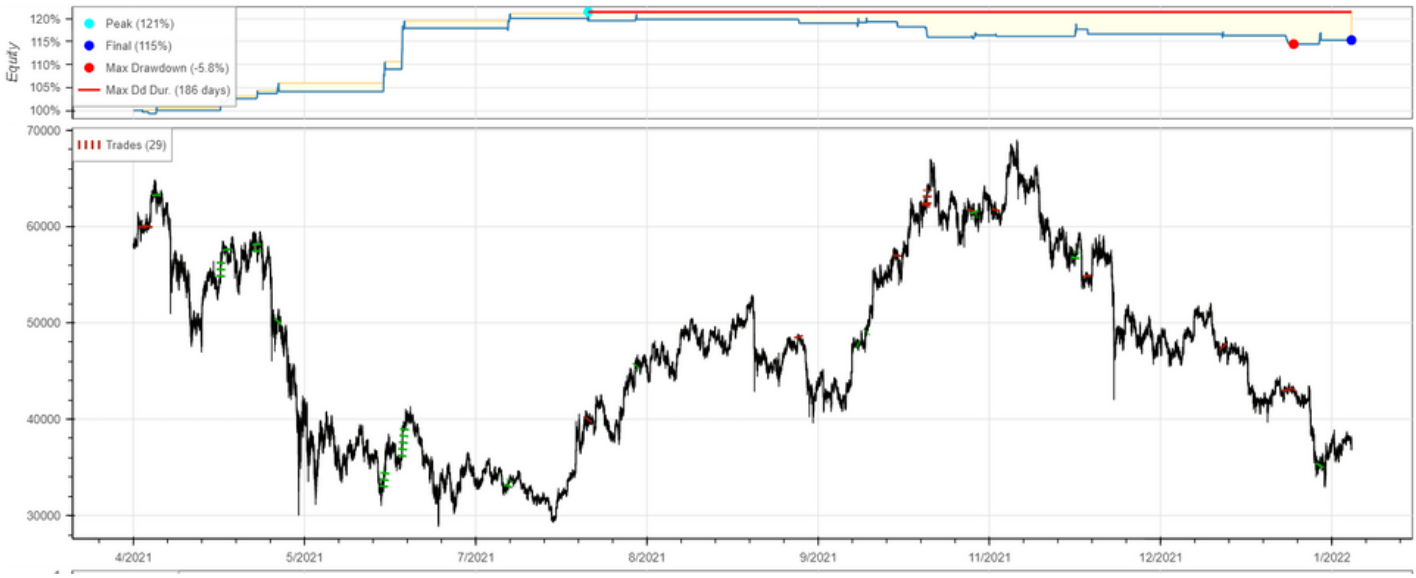
Despite its widespread use, MACD is not without limitations. Being a lagging indicator, it is inherently based on historical data, which means signals may occur after a significant price movement. This delay can sometimes lead to missed opportunities or false signals in highly volatile markets. Additionally, MACD may not be as effective in sideways, trendless markets.

MACD is a versatile tool used across various financial instruments, including stocks, commodities, and currencies. Its ability to provide insights into both the trend and momentum of the market makes it a popular choice for both short-term traders and long-term investors. By combining MACD analysis with other technical indicators and fundamental analysis, traders can enhance their decision-making process, striving for more informed and strategic market entries and exits

Final Strategy and PnL chart:

- Use **Random Forest Classifier** to predict whether the price will go up or go down in the next tick

- Use **Moving Average Convergence Divergence crossover** to verify trades



- Go long if **MACD** line crosses over the signal line and the classifier predicts that the price would go up in the next tick

- Go short if **MACD** line crosses below the signal line and the classifier predicts that the price would go down

RESULTS:

- **Start:** The strategy started on **2021-04-09** at **08:30:00**.
- **End:** It ended on **2022-01-31** at **05:30:00**.
- **Return :** The return percentage of the strategy is **15.335159 %**.
- **Volatility (Ann.):** The annualized volatility is **11.497613 %.**
- **Sharpe Ratio:** The Sharpe ratio is **1.66076**.

Start	2021-04-09 08:30:00
End	2022-01-31 05:30:00
Duration	296 days 21:00:00
Exposure Time [%]	3.00773
Equity Final [$]	115335.159225
Equity Peak [$]	121470.326645
Return [%]	15.335159
Buy & Hold Return [%]	-36.485558
Return (Ann.) [%]	19.094776
Volatility (Ann.) [%]	11.497613
Sharpe Ratio	1.66076
Sortino Ratio	5.722573
Calmar Ratio	3.307869
Max. Drawdown [%]	-5.772531
Avg. Drawdown [%]	-1.003115
Max. Drawdown Duration	186 days 00:00:00
Avg. Drawdown Duration	15 days 12:00:00
# Trades	29
Win Rate [%]	48.275862
Best Trade [%]	8.237791
Worst Trade [%]	-2.425372
Avg. Trade [%]	0.533899
Max. Trade Duration	0 days 16:00:00
Avg. Trade Duration	0 days 07:00:00
Profit Factor	2.429244
Expectancy [%]	0.554387
SQN	1.391674
_strategy	MLStrategy(lb_fa...
_equity_curve	...
_trades	Size EntryB...
dtype: object	

- **Sortino Ratio:** The Sortino ratio is **5.722573**.
- **Calmar Ratio:** The Calmar ratio is **3.307869.**
- **Max. Drawdown :** The maximum drawdown is **-5.772531%**.
- **# Trades:** The number of trades executed is **29**.
- **Win Rate:** The Win Rate is **48.275862%.**

SUGGESTED IMPROVEMENTS:

1. **Enhance Model Accuracy through Advanced Algorithms and Fine-Tuning:** To improve the accuracy of financial forecasting models, exploring advanced machine learning and statistical algorithms is crucial. This could involve experimenting with ensemble methods like Random Forest or Gradient Boosting, or deep learning approaches like Neural Networks, which can capture complex patterns in data. Fine-tuning existing models is another key strategy. This includes optimizing hyperparameters, using techniques like grid search or random search, and implementing cross-validation to ensure the model generalizes well to new data. Regular updates and refinements based on the latest market data can also help maintain high accuracy levels.

2. **Optimize Risk Management with Advanced Techniques:** Effective risk management is essential in financial forecasting. This can involve implementing more sophisticated risk assessment models, such as Value at Risk (VaR) or Conditional Value at Risk (CVaR), which provide a quantitative measure of the potential loss in value of a portfolio. Incorporating stress testing and scenario analysis into the risk management process can also be valuable. These methods assess the resilience of the investment strategy under extreme market conditions, helping to ensure that the portfolio can withstand unexpected market shocks.

SUGGESTED IMPROVEMENTS:

3. **Back-Testing with Varied Market Conditions:** Extensive back-testing across a variety of market conditions, including both typical and extreme scenarios, is vital for assessing the robustness of financial models. This involves testing the model against historical data from different market phases, such as bull markets, bear markets, periods of high volatility, and financial crises. This comprehensive back-testing helps in understanding how the model would have performed under different conditions and can provide insights into potential areas of improvement.

4. **Enhanced Feature Engineering for Better Market Representation:** Feature engineering plays a critical role in the predictive power of financial models. This involves creating new input variables (features) or transforming existing ones in ways that better capture the complexities and dynamics of financial markets. Techniques such as Principal Component Analysis (PCA) for dimensionality reduction, or the incorporation of alternative data sources (like sentiment analysis from news articles or social media) can provide more informative features. Additionally, time-series specific transformations, such as lag variables or rolling window statistics, can help in capturing temporal dynamics in the data.