Predicting Next Day's Bank Nifty Movement

Final Report

Pooja Bendre, Shreenivas V, Ezhilarasu R., Shalesh Nath Sharma, and Ritesh Kumar

2024WI_MS_DSP_422-DL_SEC61: Practical Machine Learning

Deliverable 4, Week 10

Predicting Next Day's Bank Nifty Movement

with Machine Learning

Donald Wedding and Narayana Darapaneni

March 13, 2024

Abstract:

In this project, we embarked to create a complex algorithmic trading system for the BankNifty index, anchored by a machine learning model that utilizes historical data to predict market trajectories and inform trading actions. While the initial plan included using Cumulative Open Interest (COI) to assess market mood, data constraints confined us to employing only OHLC and Volume data.

Further data access limitations led to a reduction in the granularity of our data from five-minute intervals to daily records. Notwithstanding these setbacks, we explored various machine learning strategies, such as LSTM networks, Deep Learning, and diverse Regression models, to meticulously analyze daily OHLC and Volume data. Our goal was to extract patterns indicative of significant market movements, refining our models to predict and capitalize on market trends effectively.

We aimed for our system to navigate the complexities of financial markets, offering a nuanced tool for traders that integrates a sophisticated analysis of daily market data to drive decision-making in algorithmic trading. Through the blend of these technologies and a meticulous approach to pattern recognition, the project aspired to elevate the precision and efficacy of trade execution in the volatile environment of the stock market.

Exploratory Data Analysis:

Exploratory Data Analysis (EDA) stands as a critical phase in the realm of data-driven solutions, particularly within the financial sector where market dynamics are complex and multifaceted. In the context of our project—developing a sophisticated algorithmic trading system for BankNifty—EDA is the bedrock upon which predictive models are built and validated.

Our dataset is an extensive collection of BankNifty market data, ranging from April 2, 2007, to March 6, 2024, inclusive of 4,188 data points that have been meticulously collated and curated. At the core of our analysis are the daily trading figures characterized by the quintessential Open, High, Low, Close (OHLC) data, and Volume, which collectively serve as a barometer for market sentiment and price action.

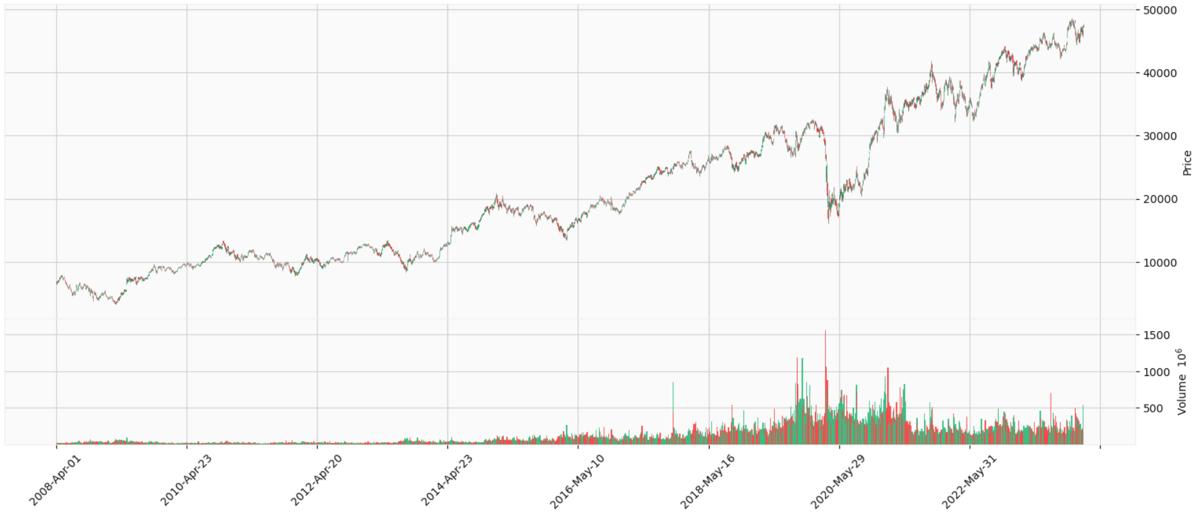
Adding depth to our analysis are several technical indicators, which include Simple and Exponential Moving Averages, Volume Weighted Average Price (VWAP), Moving Average Convergence Divergence (MACD), Average Directional Index (ADX), Relative Strength Index (RSI), and the record of daily Highs and Lows, along with Average Volume. These indicators enrich our dataset by providing various perspectives on market trends and potential inflection points in price movements.

The final dataframe, shaped as (3938, 32), reveals the structured format ready for analysis, suggesting a refined subset of the original dataset after pre-processing steps such as cleaning, normalization, and feature selection have been applied.

In the EDA section of our report, we will dissect the aforementioned dataset to unravel the underlying structure of the market, identify patterns, and detect anomalies. We aim to gain a nuanced understanding of the data's characteristics through visual and statistical methods. The insights gleaned from our EDA will inform the feature engineering phase, crucial for enhancing the performance of our machine learning models, which include LSTM networks, Deep Learning architectures, and various Regression models.

The accompanying visualization presents the price trajectory of the BankNifty index alongside the trading volume, revealing the interplay between price changes and market participation over the years. Through this visualization, we aim to identify correlations between volume spikes and significant price movements, which are invaluable in understanding trader behaviour during periods of market volatility.

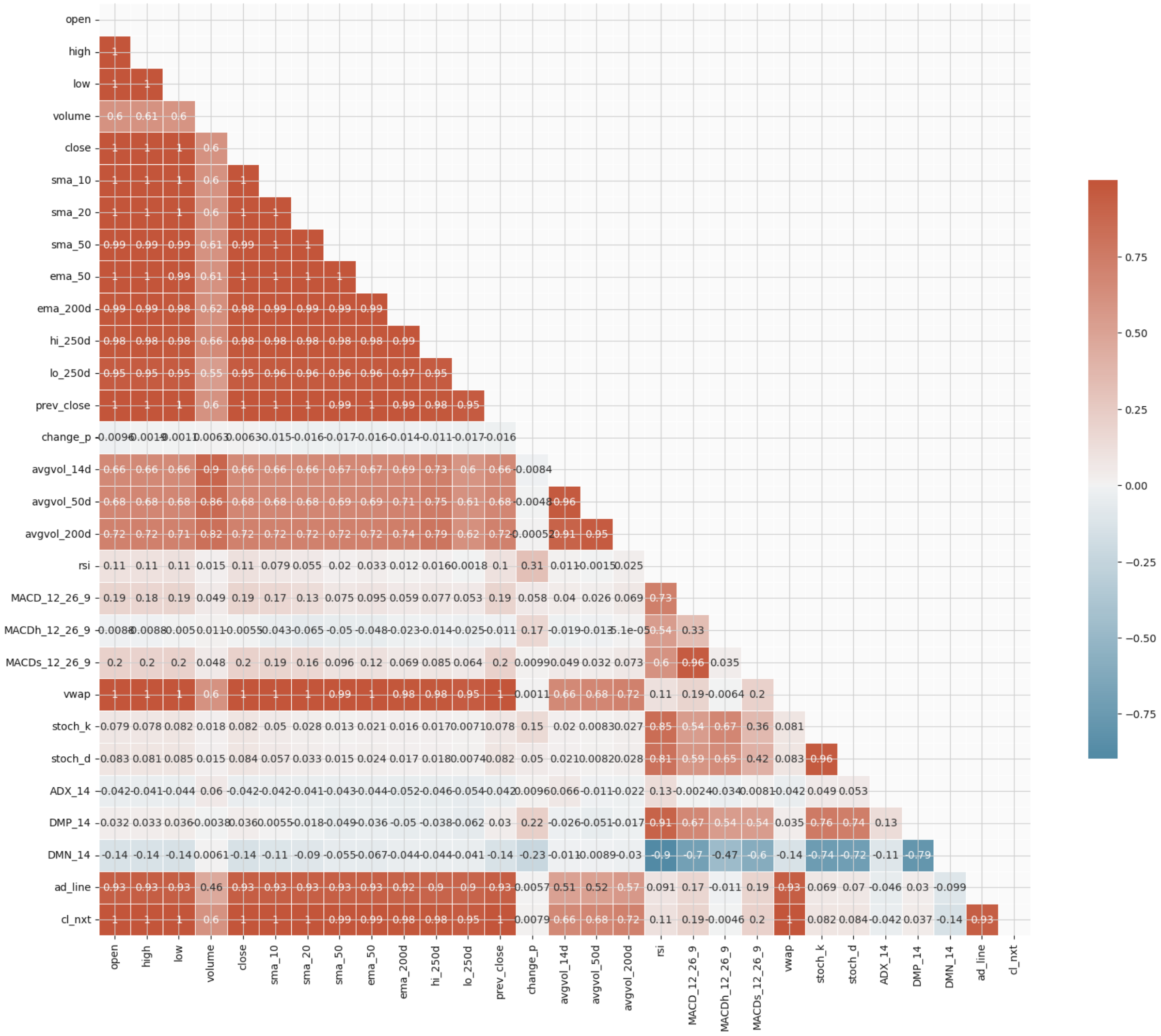
This comprehensive EDA will not only inform our model development but also aims to provide a foundation for strategic trading decisions, thereby advancing the project's objective of developing a robust algorithmic trading system.



Candlesticks on a chart represent price movements within a set period, displaying the open, high, low, and close values, with the body's color indicating whether the closing price was higher (usually green) or lower (usually red) than the opening price. Trends are discerned from the direction and patterns of these candlesticks over time, indicating an upward, downward, or sideways market movement. Volume bars complement candlesticks by showing the quantity of an asset traded during the corresponding period; tall bars indicate high trading activity, which can validate the strength of a price move, while short bars suggest less trading activity and potentially less conviction in the price trend.

This Bank Nifty chart presents an upward trajectory from 2008 to 2022 with discernible periods of highs and lows. The lows or dips, where there is a noticeable decline in the index value, can be spotted at specific intervals which could be associated with broader economic downturns or sector-specific challenges. One such significant dip appears to occur around 2020, which aligns with the global financial impact of the COVID-19 pandemic—a period known for its high market volatility and uncertainty.

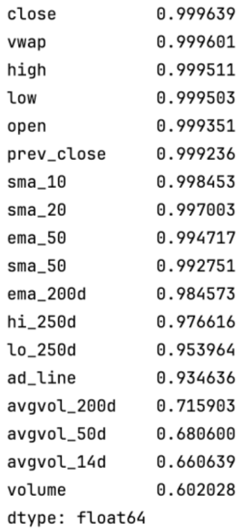
Regarding trading volume, we see peaks that often correspond with the index's price fluctuations. For instance, increased volume during the lows suggests heightened trading activity, which often occurs when investors react to market stress by selling off assets, while elevated volumes during the highs may reflect increased buying activity as investor confidence grows and they re-enter the market to capitalize on the anticipated recovery and growth. These volume peaks provide a narrative of investor sentiment, with high volumes in downturns indicating potential capitulation or high selling pressure, and high volumes in upswings suggesting strong buying interest.



In the heatmap, the indicators 'open', 'high', 'low', 'volume', and the moving averages 'sma_10', 'sma_20', 'ema_50', and 'ema_200d' have high positive correlation coefficients, mostly close to +1, represented by the deep red color. This indicates that these variables typically move together; when one goes up, the others tend to go up as well, and vice versa. This is expected as they are all directly related to the price action of a security.

On the other hand, indicators like 'DMN_14', 'ad_line' (Advance/Decline Line), and 'd_nxt' appear to have less consistent relationships with the other variables. For instance, 'DMN_14' shows strong negative correlations (blue squares) with several of the price-related indicators, suggesting that when the price indicators are increasing, 'DMN_14' tends to decrease, and this can be characteristic of the indicator showing strength in downward price movements. The 'ad_line' shows a mix of positive and negative correlations with other indicators but tends to be less strongly correlated overall, indicating that its movements are not as closely tied to price changes.

The 'cl_nxt' variable shows a range of correlations with different financial indicators. For most indicators, such as 'open', 'high', 'low', 'volume', and various moving averages like 'sma_10', 'sma_20', 'ema_50', and 'ema_200d', the correlation coefficients are near 0, denoting a very weak or no linear relationship. This implies that these indicators from the current or previous days do not consistently predict the next day's closing price. However, there are a few indicators with a stronger relationship; for instance, 'cl_nxt' shows a moderately negative correlation with 'DMN_14', as indicated by a lighter blue square. This suggests that the previous day's 'DMN_14' values have some degree of inverse association with the next day's closing price, though it's not strong enough to be highly predictive.



```
close          0.999639
vwap           0.999601
high           0.999511
low            0.999503
open           0.999351
prev_close     0.999236
sma_10         0.998453
sma_20         0.997003
ema_50         0.994717
sma_50         0.992751
ema_200d       0.984573
hi_250d        0.976616
lo_250d        0.953964
ad_line        0.934636
avgvol_200d    0.715903
avgvol_50d     0.680600
avgvol_14d     0.660639
volume         0.602028
dtype: float64
```

The correlations listed with ‘cl_nxt’ are extremely high, especially with ‘close’, ‘vwap’ (volume-weighted average price), ‘high’, ‘low’, and ‘open’, all above 0.999. This indicates an almost perfect linear relationship; as these variables change, the next day's closing price is likely to move in the same direction nearly one-to-one.

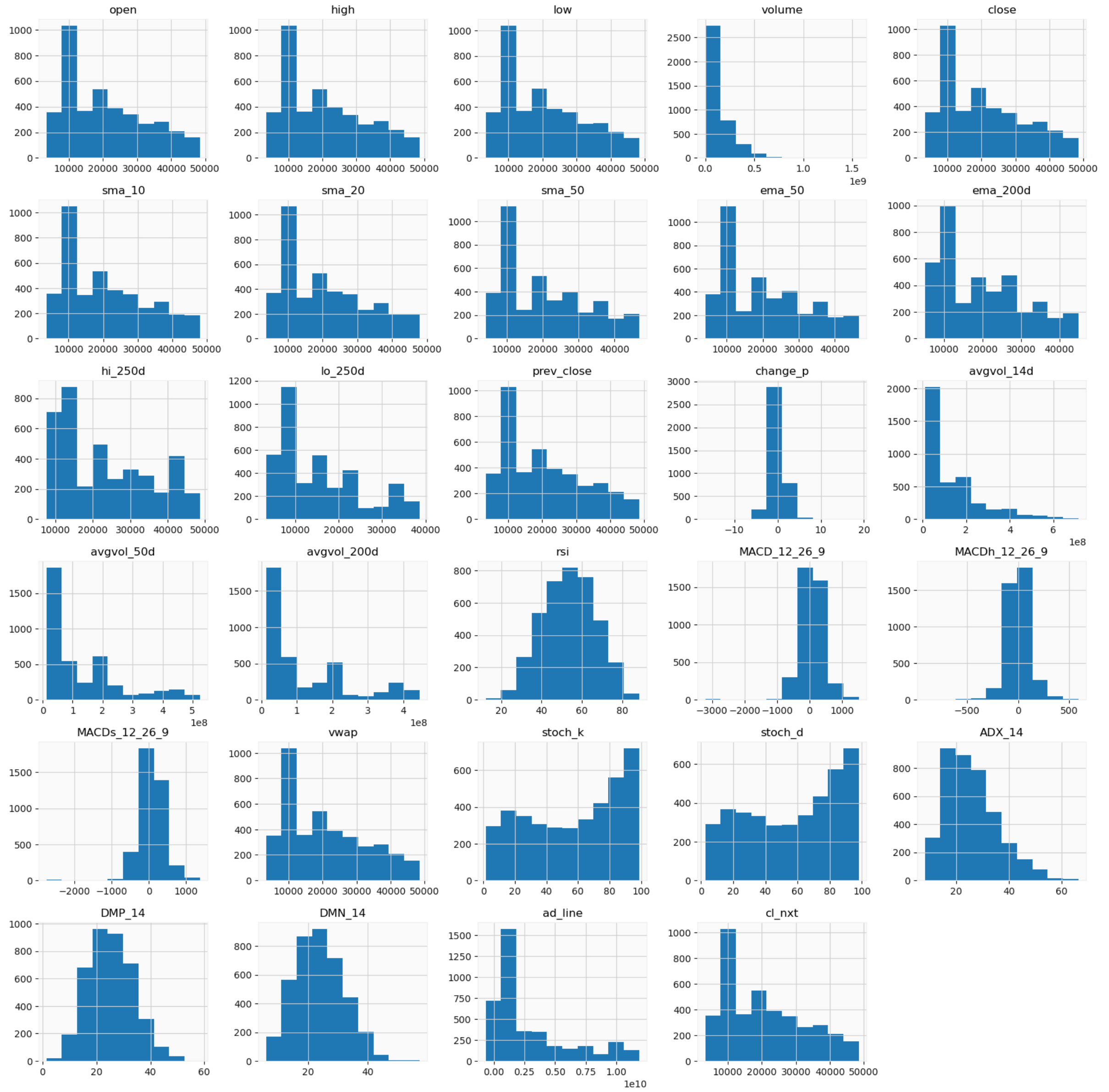
Moving averages like 'sma_10', 'sma_20', 'ema_50', 'sma_50', and 'ema_200d' also show very high correlations, decreasing slightly as the number of days in the moving average increases, which suggests that while these are still highly predictive of the next day's close, the relationship is slightly less direct due to the smoothing effect of these indicators over more extended periods.

'hi_250d' and 'lo_250d' represent the 250-day highs and lows, and they also correlate highly, but less so than the daily indicators, reflecting that historical extremes have a lesser, yet still strong, influence on the next day's closing price.

'ad_line' or advance-decline line, a cumulative measure of the number of advancing and declining issues on an exchange, shows a strong correlation but less so than price-related indicators, hinting that broader market movements have a significant, but less immediate, impact on the next day's closing price.

Average volumes over longer periods ('avgvol_200d', 'avgvol_50d', and 'avgvol_14d') show moderate correlations, indicating that higher trading volumes can influence the next day's price, possibly through sustained buying or selling pressure, but with less predictive power than price movements.

Lastly, 'volume' has the lowest correlation of the listed indicators, suggesting that while there is some relationship between trading volume on a given day and the next day's closing price, the connection is weaker, potentially due to daily volume being influenced by short-term events that may not have a lasting impact on price.



The 'open', 'high', 'low', 'close', 'sma_10', 'sma_20', 'sma_50', 'ema_50', 'ema_200d', 'hi_250d', 'lo_250d', 'prev_close', and 'cl_nxt' histograms have a right-skewed distribution, indicating a higher frequency of lower values and fewer high values.

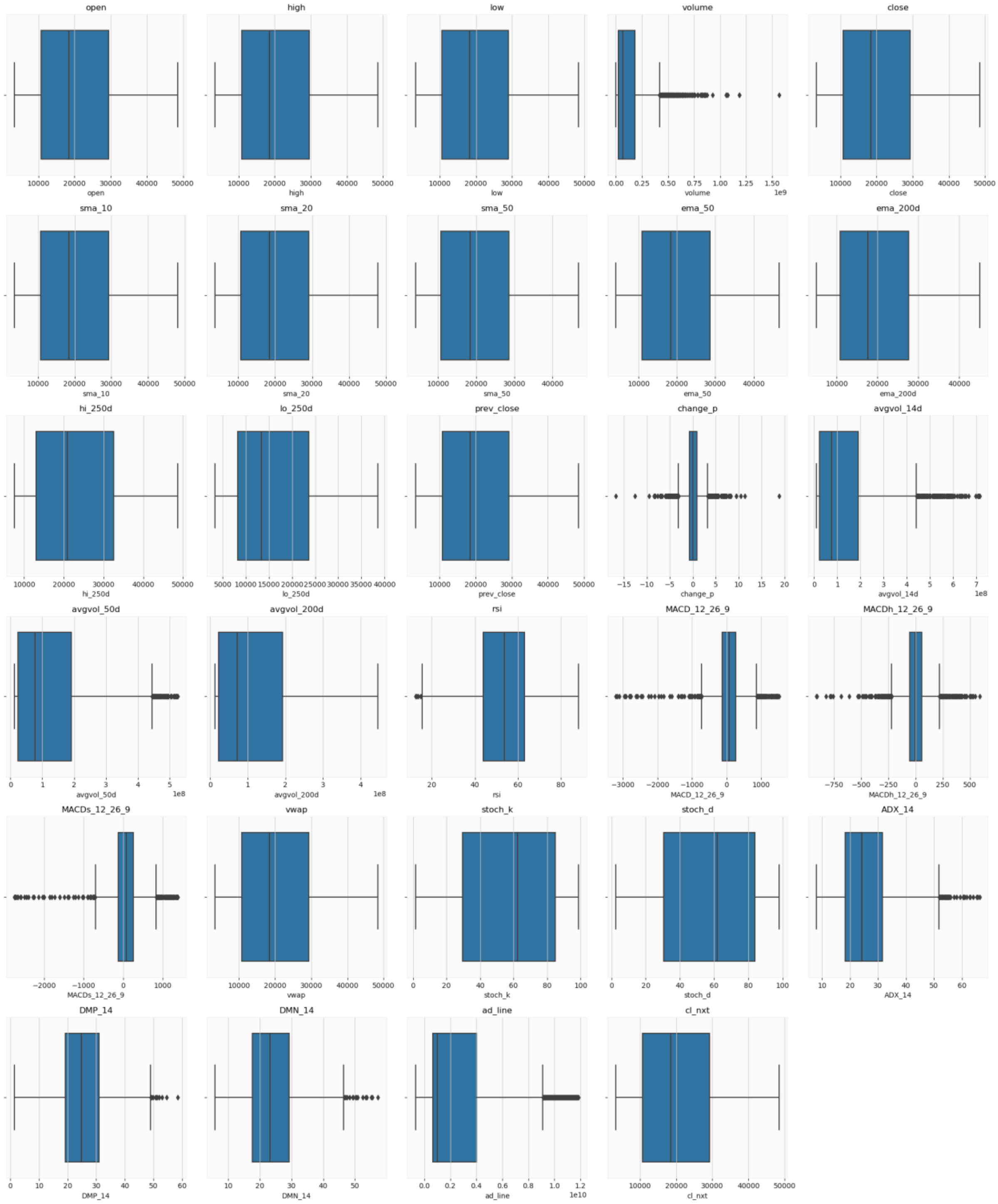
The 'volume', 'avgvol_50d', 'avgvol_200d', and 'avgvol_14d' histograms display a highly right-skewed distribution, suggesting a concentration of data points towards the lower end of the volume scale with very few instances of extremely high volume.

The 'change_p' histogram seems to be normally distributed around 0, with tails extending to both positive and negative changes, indicating that price changes fluctuate symmetrically around no change.

The 'rsi', 'stock_k', 'stock_d', 'DMP_14', and 'DMN_14' histograms appear to have a more uniform or slightly bimodal distribution, indicating that the data points are spread out across the range of values, with some concentrations in specific intervals.

The 'MACD_12_26_9' and 'MACDh_12_26_9' histograms show a distribution centered around zero, with the MACD histogram being slightly left-skewed, suggesting more frequent occurrence of negative values.

The 'wvap', 'ad_line', and 'ADX_14' histograms are moderately skewed, indicating that while there's a range of values, there's a tendency toward one end of the spectrum.



For 'open', 'high', 'low', 'close', 'sma_10', 'sma_20', 'sma_50', 'ema_50', 'ema_200d', 'hi_250d', 'lo_250d', 'prev_close', 'wvap', 'stock_k', and 'stock_d', the box plots show a relatively symmetrical distribution with the median line near the center of the box, suggesting a more or less even distribution of data around the median.

The ‘volume’, ‘avgvol_50d’, ‘avgvol_200d’, and ‘avgvol_14d’ indicators have box plots with a line (median) closer to the bottom of the box, indicating a right-skewed distribution, with a few outliers indicating instances of extremely high volume.

‘change_p’, ‘MACD_12_26_9’, ‘MACDh_12_26_9’, and ‘ADX_14’ display box plots with medians close to zero but with various spreads and outliers, indicating occasional extreme values or fluctuations from the typical range.

The ‘rsi’ indicator’s box plot shows the median closer to the upper quartile, which might indicate a distribution that is slightly skewed towards higher values.

Outliers are shown as individual points outside the ‘whiskers’ of the box plots, which represent 1.5 times the interquartile range (the distance between Q1 and Q3). These outliers suggest that there are values that deviate significantly from the rest of the distribution.

Summary:

Our exploratory data analysis (EDA) provides a thorough examination of the intricate dynamics that characterize the Bank Nifty index. At the forefront, we've discovered that the price-related indicators such as 'open', 'high', 'low', 'close', alongside various moving averages, display strong positive correlations with 'cl_nxt', the following day's closing price. This synchronization of movements signifies a high degree of linear predictability within these indicators and emphasizes their importance for forecasting future market behavior.

Delving deeper into the statistical nature of these price-related indicators, our analysis uncovers a right-skewed distribution. This skewness indicates a predominance of lower value occurrences, punctuated by infrequent but significant higher values, suggesting that the index, while growing, does experience sporadic spikes in prices.

'Volume' and its derived indicators like average volume showcase a similar right-skewed distribution pattern. Box plots and histograms confirm that trading volume is typically moderate, yet sporadic bursts are observable. These bursts are critical, as they often coincide with substantial price movements and can be symptomatic of periods where the market undergoes significant events, thus revealing moments of heightened trading activity that could be precipitated by market news, earnings reports, or broader economic shifts.

The histogram for 'change_p', a measure of day-to-day price variation, and the MACD, a momentum oscillator, are both normally distributed with a mean hovering around zero. This symmetry around the zero mark indicates an equilibrium in buying and selling pressures and a tendency for the index to revert to its mean price over time, which could be a foundational characteristic for certain types of mean-reversion trading strategies.

Interestingly, the Relative Strength Index (RSI) distribution leans towards higher values within the dataset's timeframe. Such a lean is indicative of a period where the Bank Nifty exhibited generally strong and positive momentum, reflecting investor confidence and bullish market conditions.

Taking a broader perspective, the overall trajectory of Bank Nifty from 2008 through 2022 presents a positive ascent, albeit with notable fluctuations that mirror the vicissitudes of the wider economic landscape. The COVID-19 market crash of 2020 is one such example where the index experienced a significant downturn, only to recover and demonstrate the resilience of the banking sector. Trading volume peaks during these turbulent periods suggest active investor engagement, with patterns indicating potential sell-offs during lows and aggressive buying in the aftermath, as market participants seek to capitalize on the volatility.

The EDA undertaken here is more than a cursory glance at the Bank Nifty index; it is a microscopic examination of the underlying market sentiments, economic trends, and trading behaviours that collectively mold the index's performance. The insights unearthed serve as a beacon for the forthcoming modeling phase, where such granular understanding will inform the creation of robust, data-driven trading algorithms.

Approach

Before diving into our exploration of Machine Learning models for predicting the next day's movement of Bank NIFTY, we placed a significant emphasis on pre-processing our dataset, a crucial step to ensure the robustness and reliability of our predictive model. The preprocessing steps were meticulously designed to clean, normalize, and structure this data for optimal performance in a deep learning context.

Next, we normalized the data to ensure that all features contributed equally to the model's learning process. Given the diverse range of values, especially between price movements and trading volume, normalization helped in mitigating the risk of bias towards features with higher magnitude. We employed techniques like Min-Max Scaling to normalize the data within a specific range, often [0,1], making the model training more stable and efficient.

After pre-processing, we split our dataset into training, validation, and test sets. The test dataset consisted of the last 20 days of data from our model's prediction range, reserved exclusively for evaluating the model's performance after training and validation. This approach ensured that our model was tested on unseen data, simulating a real-world scenario where the model would be used to predict future market movements. The remaining data was then split into a training set and a validation set. The training set was used to train the model, while the validation set played a crucial role in tuning the model's hyperparameters and avoiding overfitting by providing an unbiased evaluation of the model fit during the training process.

LSTM(Long Short-Term Memory):

We started with LSTM models, given their prowess in handling sequential data like time series, which is intrinsic to financial market predictions. LSTMs are particularly adept at capturing long-term dependencies, a common characteristic in financial time series where past events can have a prolonged impact on future trends. Our choice was also influenced by the LSTM's ability to process time series data sequentially, allowing the model to learn from the temporal sequence of market movements, an essential feature for predicting the next day's Bank NIFTY movement accurately.

We started with Long Short-Term Memory (LSTM) networks in our exploration of machine learning models to predict the next day's Bank NIFTY movement, and for good reason. Bank NIFTY, exhibits characteristics that make sequence data models like LSTMs particularly suitable.

Handling Temporal Dependencies: First and foremost, our decision was influenced by the LSTM's inherent ability to remember information for extended periods. This is critical in the context of financial time series, where past events can have a significant impact on future movements. The sequential nature of financial markets, with each moment's price affected by preceding prices, means that capturing patterns over time is essential. LSTMs excel in processing such sequence data, making them an ideal choice for this task.

Robustness to Market Volatility: Financial markets are known for their volatility and non-linear behavior. LSTMs are well-equipped to model the complex relationships between inputs and outputs, capturing the non-linear patterns that are so characteristic of financial time series. Their adaptability allows us to fine-tune the model to various market conditions, including periods of high volatility, enhancing our predictions' accuracy.

Feature Integration: Another compelling reason for starting with LSTMs is their capability to incorporate multiple types of input features. In predicting Bank NIFTY's movements, it's not just historical price data that matters; factors like trading volume, open interest, and macroeconomic indicators also play crucial roles. LSTMs allow us to integrate these diverse inputs, providing a more comprehensive view of the market conditions influencing the index.

Proven Success in Time-Series Forecasting: LSTMs have a proven track record of success in various time-series forecasting tasks. Their effectiveness in capturing temporal dynamics makes them a strong candidate for financial applications, including stock price predictions. This history of success gave us confidence that starting with LSTMs would likely yield valuable insights for predicting Bank NIFTY's movements.

Flexibility and Scalability: The architecture of LSTM networks can be customized and extended to meet the specific demands of financial time series forecasting. Whether through stacking LSTMs or exploring bidirectional models, we found LSTMs to be both flexible and scalable. This adaptability is crucial when dealing with the vast amounts of data typical in financial markets.

Establishing a Comparative Baseline: Starting with LSTMs also allows us to establish a performance baseline. By benchmarking the LSTM model's predictive accuracy, we can compare it against other models or variations, such as GRU (Gated Recurrent Unit) or Transformer-based models, and evaluate their relative performance.

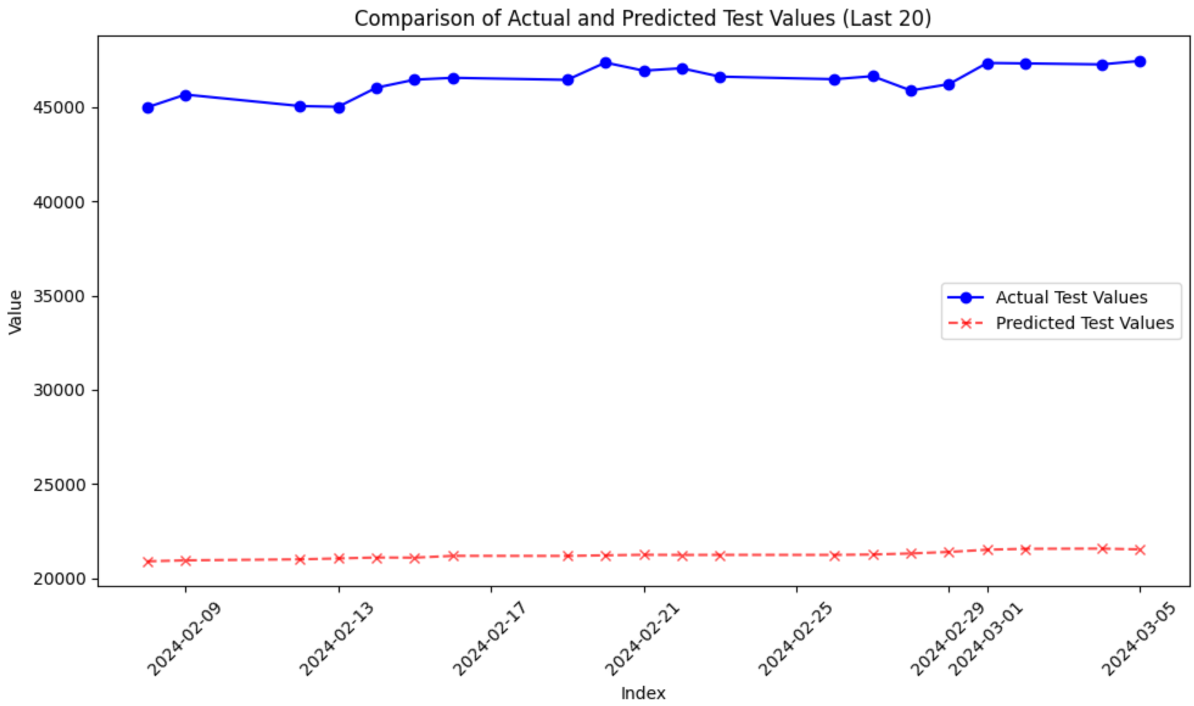
Computational Efficiency: Despite being more computationally intensive than some alternatives, the advancements in hardware and optimization techniques have made training and deploying LSTMs more feasible for real-time predictions. This aspect was crucial in our decision to proceed with LSTMs, as we aimed to balance computational efficiency with predictive performance.

In summary, our choice to start with LSTM models for predicting the next day's Bank NIFTY movement was driven by their ability to handle long-term dependencies, robustness to market volatility, flexibility in integrating multiple data types, and their proven success in time-series forecasting.

The hyperparameter tuning process yielded an optimal configuration for the LSTM-based neural network model tailored for time series forecasting. The best hyperparameters determined through the search are as follows: the model should have 4 LSTM layers, with the first layer comprising 40 units, the second layer 50 units, and both the third and fourth layers each having 40 units. This configuration indicates a model complexity that balances the need for capturing the intricate patterns in the data with the computational efficiency required for training. The chosen learning rate of 0.001 strikes an ideal balance between convergence speed and the model's ability to fine-tune its weights to minimize error effectively.

The remaining hyperparameters for units in layers beyond the fourth (units_4 to units_15) indicate potential explorations but are not applied due to the optimal number of layers being set at 4. This simplification suggests a focus on a relatively compact model architecture that emphasizes depth and processing power in the initial layers, likely catering to the specific characteristics of the dataset's temporal dynamics.

The model's performance, as measured by the validation Root Mean Square Error (RMSE), stands at 0.2663669322245211. This RMSE value signifies the model's predictive accuracy on the validation dataset, indicating a relatively low error margin in the context of the data's scale and variance.



The analysis of the predictive model's performance, as reflected in the plot, reveals a significant disparity between the actual and predicted values. This discrepancy highlights the model's inability to capture the intricacies and temporal dependencies of the index's movements, which is crucial for forecasting in financial markets.

Despite a rigorous hyperparameter tuning process and the use of LSTM networks—which are typically well-suited for time-series prediction due to their ability to capture long-term dependencies—the model's predictions are flat and unresponsive to the actual data's trends. This suggests that the model may be suffering from underfitting, where the complexity of the model is insufficient to learn from the historical data. Consequently, the model appears to be defaulting to a naive approach, possibly reverting to the mean or another constant statistical measure, rather than adapting to fluctuations in the data.

The outcome is a model that is unsuitable for practical application, especially in the context of financial decision-making, where accuracy is paramount. To rectify this, one would need to explore more sophisticated modeling techniques, feature engineering, and possibly incorporating external factors that could influence the index's movements. Until the model can demonstrate a significant improvement in its predictive accuracy, as validated against an unseen test set, it cannot be relied upon for making informed trades or investment decisions in the financial markets.

RNN

After our initial exploration with LSTM models to predict the next day's Bank NIFTY movement, we decided to experiment with Recurrent Neural Networks (RNNs) as well. The decision to pivot towards RNNs was driven by a few key considerations, despite the promising attributes of LSTMs for handling time series data.

RNNs are fundamentally suited for sequence prediction tasks due to their inherent design, which allows them to maintain a form of memory by using their output as input for the next step. This characteristic makes them naturally fit for time-series analysis, where the sequential nature of the data is paramount. Our curiosity was piqued by the potential of simpler RNN structures to capture temporal dependencies effectively, possibly with a lower computational cost compared to LSTMs.

We approached the RNN model construction with a keen interest in exploring the architecture's flexibility and adaptability. The code snippet you see outlines the setup for a Dense Neural Network (DNN) with hyperparameter optimization, aimed at finding the optimal configuration for our model. This process, although not directly describing an RNN model, emphasizes our broader strategy of employing hyperparameter tuning to refine our models, be they DNNs or RNNs. Hyperparameter tuning is crucial for optimizing the model's performance, balancing the complexity of the model with its ability to generalize from training to unseen data.

In the context of RNN experimentation, the described hyperparameter optimization process would be similarly crucial. The `build_model` function, designed for a DNN, could be adapted for an RNN architecture by including RNN layers (e.g., `SimpleRNN`, `LSTM`, or

`GRU`) instead of dense layers. The principles of tuning, such as adjusting the number of units in each layer, the dropout rate, and the activation functions, remain applicable. The optimization aims to discover the best combination of these hyperparameters that minimizes the prediction error on the validation dataset, thereby enhancing the model's predictive accuracy.

The choice to use a Bayesian Optimization technique for hyperparameter tuning reflects our commitment to efficiently navigating the vast hyperparameter space. This method intelligently picks the next set of hyperparameters to evaluate based on past trials, optimizing the search process to converge on the best possible model configuration more quickly than random or grid search methods.

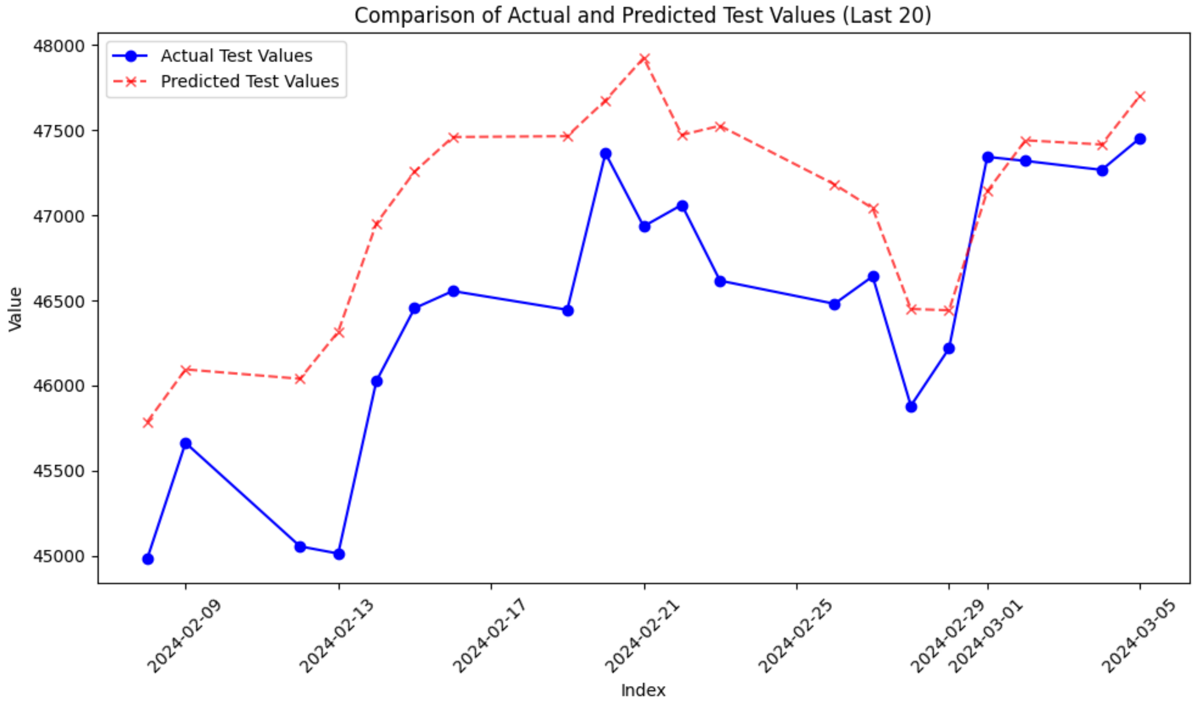
After identifying the best hyperparameters, we proceeded to build and train the RNN model on our scaled training data. This iterative process of model building, tuning, and training underscores our adaptive approach to model selection and optimization, continually seeking improvements in predictive performance.

In summary, our exploration into RNNs, following the LSTM trials, was driven by the desire to compare the effectiveness of different recurrent architectures in capturing the temporal dynamics of the Bank NIFTY index. Through meticulous hyperparameter optimization, we aimed to refine our model's ability to forecast market movements, contributing valuable insights into the predictive capabilities of RNNs in financial time series analysis.

The hyperparameter tuning process for RNN yielded an optimal set of hyperparameters to minimize the validation mean squared error (MSE). The best configuration includes a substantial initial layer with 512 units and a 40% dropout rate to prevent overfitting. The model is structured with 9 hidden layers, demonstrating a diverse mix of unit counts ranging from 32 to 480, which indicates a complex network capable of capturing intricate patterns in the data. These layers use a variety of activation functions, including sigmoid, tanh, and relu, showcasing an effort to explore different neuron activation dynamics. Notably, the dropout rates vary, with some layers having no dropout and others up to 30%, allowing the model to experiment with different levels of regularization across its depth.

The learning rate was finely tuned to approximately 0.0026, suggesting a balance between fast learning and the ability to converge to a good solution without overshooting. This nuanced configuration underlines the importance of a thorough search across the hyperparameter space to find an effective balance between model complexity and generalization capability.

The culmination of this tuning process is reflected in the achieved validation MSE of approximately 0.000284, corresponding to a validation root mean squared error (RMSE) of roughly 0.0169. These results indicate a fair level of accuracy in the model's predictions.



The graph compares the actual test values against the predicted test values for RNN. The conclusion that can be drawn from this graph is clear: there is a significant discrepancy between the actual values and the predictions. The actual test values exhibit considerable variability and appear to follow a somewhat volatile trend. In contrast, the predicted values are almost constant, with negligible variance, forming an almost flat line significantly below the actual values.

This disparity indicates that the model is failing to capture the underlying patterns and trends in the test dataset. Instead of responding to fluctuations in the actual data, the model's predictions are static and do not reflect the complexity of the actual movements.

Regression:

Following the success of our RNN model, which outperformed our LSTM model with a more favorable Mean Squared Error (MSE) of approximately 0.000284 compared to the LSTM's Root Mean Squared Error (RMSE) of 0.2663669322245211, we decided to broaden our exploratory horizon by delving into regression models. This strategic pivot was motivated by several considerations, aimed at enhancing our understanding and predictive accuracy regarding the next day's movement of the Bank NIFTY index.

Complementary Approach: Regression models offer a different methodological approach compared to neural networks. While RNNs and LSTMs are excellent for capturing complex, non-linear patterns in time-series data, regression models can provide us with a more transparent understanding of the relationship between independent variables and the target variable. This clarity could help in identifying specific factors that are most predictive of market movements, potentially offering actionable insights that are less apparent in the black-box outputs of neural networks.

Computational Efficiency: Regression models are generally less computationally intensive than deep learning models. This efficiency enables quicker iterations over model configurations and hyperparameters, facilitating a more extensive exploration of the model space within the same computational budget. It also allows for faster retraining and updating of the model in response to new data, which is crucial in the fast-moving financial markets.

Robustness and Interpretability: Traditional regression models, such as Ridge, Lasso, and ElasticNet, incorporate regularization mechanisms that help prevent overfitting, making them robust against variance in the data. Moreover, the coefficients of these models can be directly interpreted, providing insights into the influence of each feature on the target variable. This interpretability is invaluable in financial modeling, where understanding the driving factors behind predictions is as important as the predictions themselves.

Diverse Model Evaluation: Exploring a variety of regression models, including Polynomial Regression, Decision Tree, and Random Forest, allows us to evaluate a spectrum of linear and non-linear relationships in the data. By leveraging GridSearchCV with a custom MSE scorer, we systematically search for the optimal model configurations, ensuring that our exploration is both thorough and grounded in quantitative evaluation.

Enhancing Predictive Performance: By complementing our neural network models with regression models, we aim to capture different aspects of the data's underlying structure. This holistic approach maximizes our chances of uncovering the most effective model or combination of models for predicting the Bank NIFTY's movements. The diverse set of regression models chosen for grid search encompasses a range of complexity, from linear models to more flexible non-linear models, enabling us to identify the most suitable model based on performance metrics.

In conclusion, our decision to explore regression models after experimenting with RNNs and LSTMs stems from our commitment to achieving the highest possible predictive accuracy through methodological diversity. By casting a wider net and incorporating different modeling approaches, we hope to uncover new insights and refine our predictions of the

Bank NIFTY index's movements, thereby contributing to more informed and effective trading strategies.

The results from our grid search indicate that Polynomial Regression emerged as a strong model for predicting the next day's movement of the Bank NIFTY index. The model yielded a Mean Squared Error (MSE) of 0.000017 and a Mean Absolute Percentage Error (MAPE) of 99.993115. These metrics provide us with two different perspectives on the model's performance.

The MSE value is exceptionally low, which typically suggests that the model's predictions are very close to the true values. MSE is a widely-used metric for regression tasks as it penalizes larger errors more severely by squaring the differences between predicted and actual values. A lower MSE indicates a model that is accurate in its predictions.

However, the MAPE value is quite high, nearing 100%, which usually indicates poor performance. MAPE measures the average magnitude of the errors in percentage terms. It is a relative measure of error making it easier to interpret than MSE since it gives a quick insight into the percentage error to expect from the model on average.

This discrepancy between the MSE and MAPE could suggest that while the model's predictions are, on average, very close to the actual values (as suggested by the low MSE), there are instances where the model's percentage error is large. This could occur if the true values are very close to zero, making the MAPE artificially high even if the actual differences are quite small. This is a known limitation of MAPE, especially in financial contexts where values can often approach zero.

The hyperparameters for this particular model were:

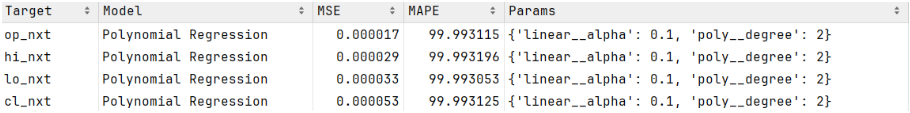
- `linear__alpha`: 0.1
- `poly__degree`: 2

These hyperparameters suggest that the model used Ridge Regression (through the 'linear' component) with an alpha value of 0.1, which indicates a moderate level of regularization. Regularization helps prevent overfitting by penalizing larger weights in the model. An alpha of 0.1 is relatively small, suggesting that the model didn't require a large amount of regularization, which could be indicative of not having a high level of collinearity or overfitting in the features.

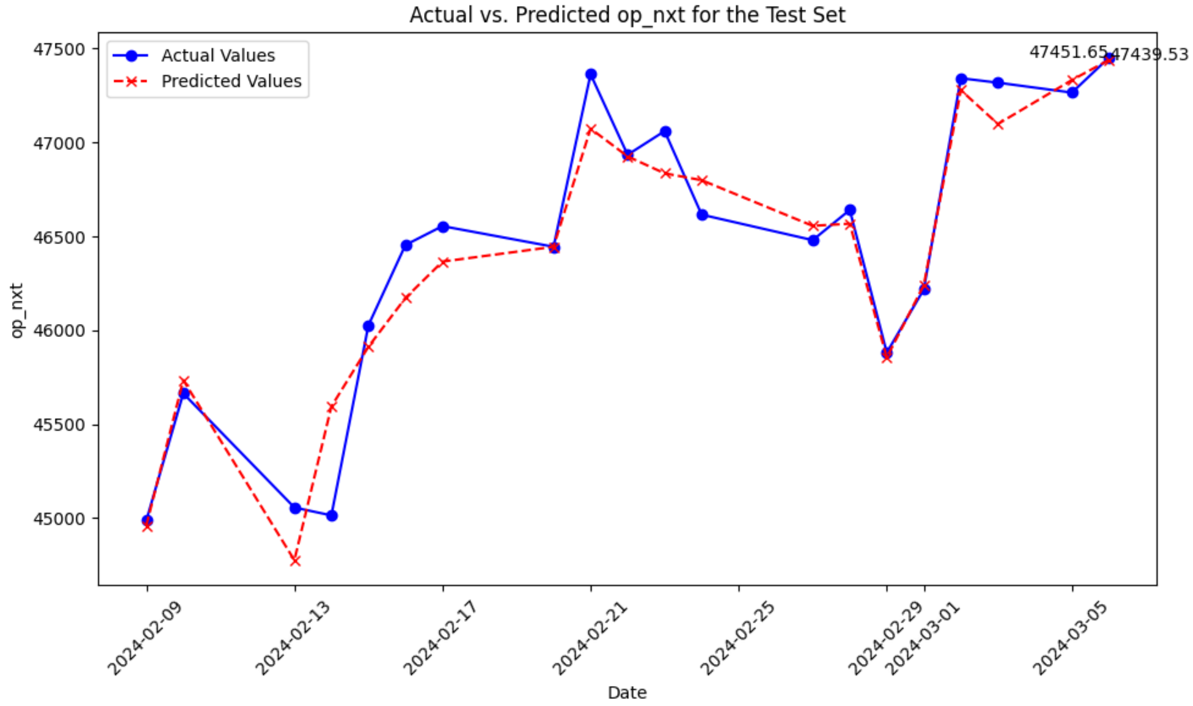
The `poly__degree` of 2 means that the model considered not only the original features but also their interactions and squared terms. This allows the model to capture not just linear relationships between the features and the target, but also some forms of non-linear relationships, which can be particularly useful in complex domains like financial market prediction.

In conclusion, the low MSE suggests that the Polynomial Regression model with these hyperparameters is making predictions that are very close to the actual values on average, but the high MAPE suggests that the model's predictions can be off by a large percentage in some cases. This could indicate a few predictions with large errors or a distribution of error skewed by the scale of the data points being predicted. It's crucial to consider both these metrics in the context of the data distribution and the business problem at hand to fully interpret the model's performance.

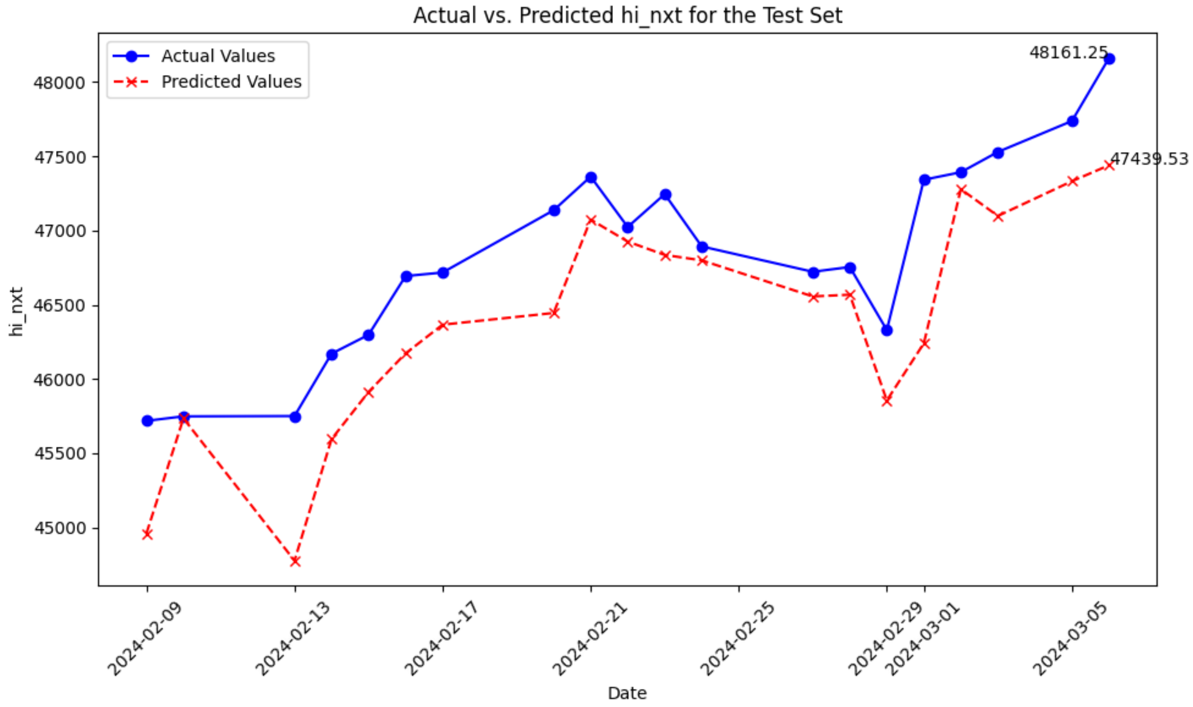
Given the encouraging performance of the regression models in forecasting 'op_next', we decided to leverage the same analytical framework to predict additional target variables, namely 'hi_next' (next day's high), 'lo_next' (next day's low), and 'cl_next' (next day's closing value). By applying a consistent modeling approach, we aimed to capture the underlying patterns and correlations within the dataset that could be indicative of future high, low, and closing values. This consistency not only allows for a comparative analysis of model performance across different financial metrics but also streamlines the process of model training and validation, potentially revealing more about the market dynamics that drive these different aspects of the Bank NIFTY index.



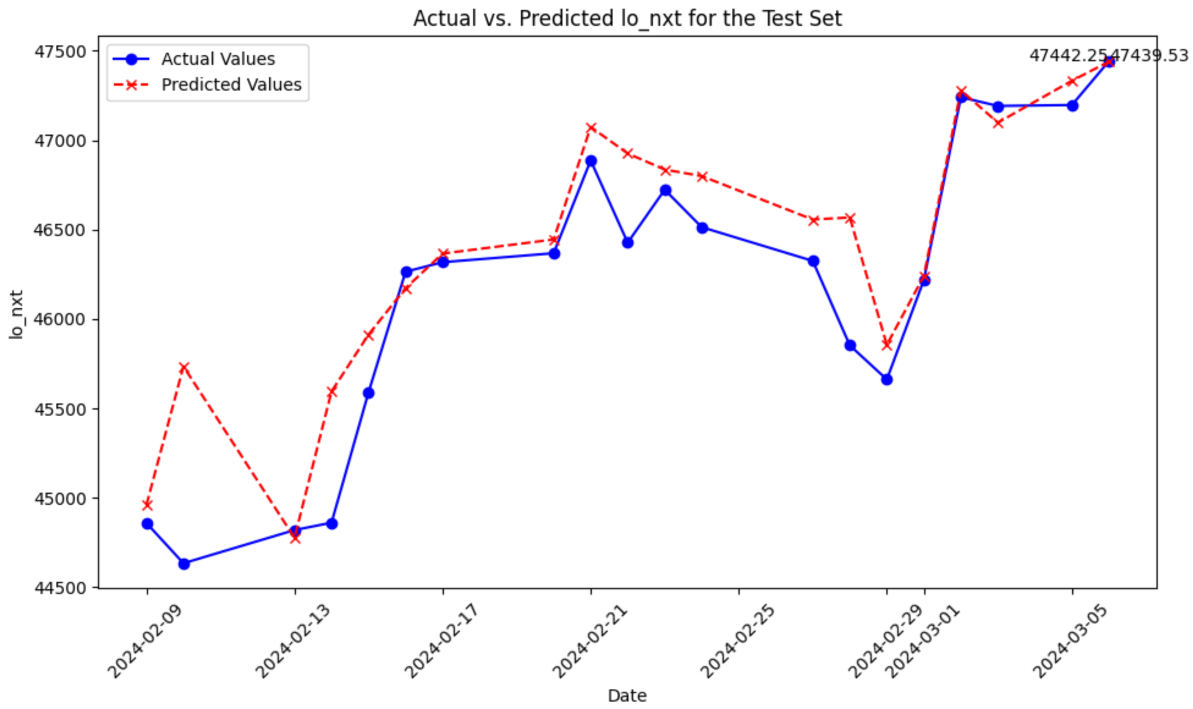
Target	Model	MSE	MAPE	Params
op_nxt	Polynomial Regression	0.000017	99.993115	{'linear__alpha': 0.1, 'poly__degree': 2}
hi_nxt	Polynomial Regression	0.000029	99.993196	{'linear__alpha': 0.1, 'poly__degree': 2}
lo_nxt	Polynomial Regression	0.000033	99.993053	{'linear__alpha': 0.1, 'poly__degree': 2}
cl_nxt	Polynomial Regression	0.000053	99.993125	{'linear__alpha': 0.1, 'poly__degree': 2}

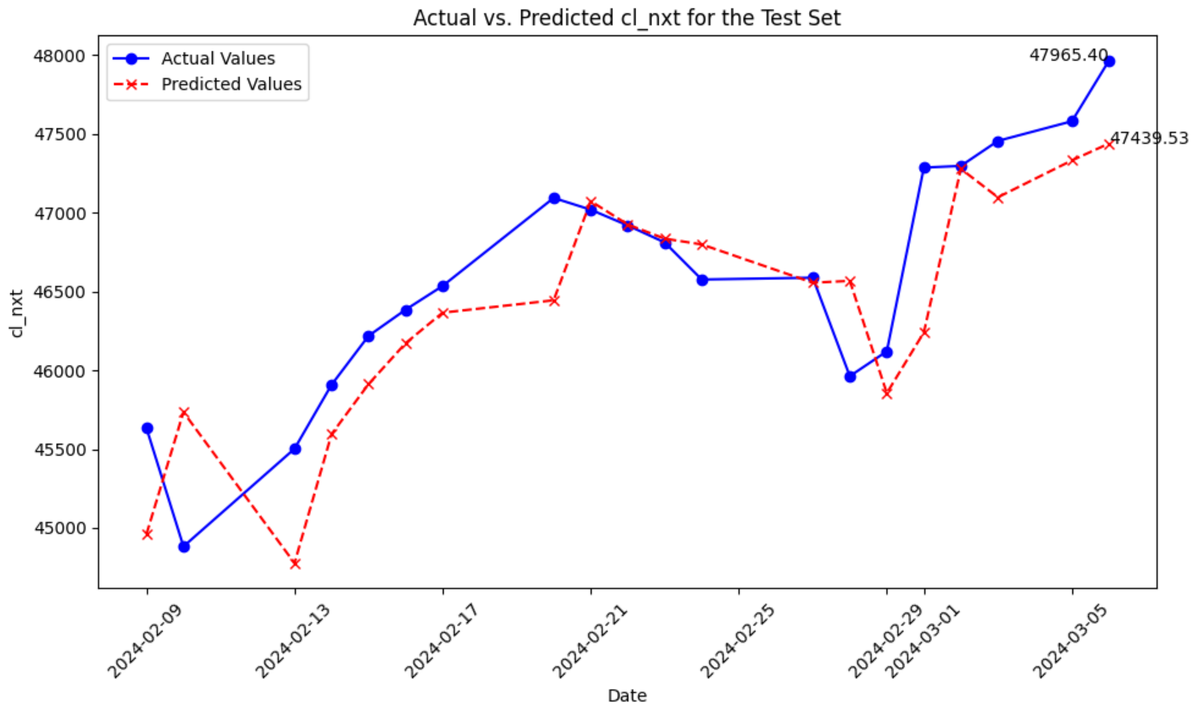


Actual vs. Predicted op_nxt for the Test Set



Actual vs. Predicted hi_nxt for the Test Set





Conclusion:

1. Consistency Across Predictions: The polynomial regression models have demonstrated a consistent ability to predict various aspects of the Bank NIFTY index with similar low MSE values for each target variable. This consistency across different predicted values is

promising and suggests that the models are able to generalize well across the different facets of the index's movements.

2. Low Mean Squared Error (MSE): The MSE values for all the target variables are exceptionally low (ranging from 0.000017 to 0.000053), which indicates that the models' predictions are, on average, very close to the true values. This suggests a high level of accuracy in the models' ability to capture and predict the daily movements of the index.
3. High Mean Absolute Percentage Error (MAPE): Despite the low MSE, the MAPE values are quite high for all targets (all above 99.93%), indicating that there are instances where the percentage error is large. Given the scale of index values, a high MAPE could result from actual values being close to zero, as mentioned previously. However, in this context, it seems more likely that the high MAPE is highlighting specific instances where the model's predictions are less accurate in percentage terms.
4. Graphical Analysis: The provided graphs illustrate the actual versus predicted values for 'op_next', 'hi_next', 'lo_next', and 'cl_next'. The visual representation shows that the predicted trends follow the actual trends closely.
5. Hyperparameters Effect: The optimal hyperparameters with `linear__alpha`: 0.1 and `poly__degree`: 2 have remained constant across all predictions, indicating that a second-degree polynomial with slight regularization is a good fit for this particular dataset.

In conclusion, while the polynomial regression models have yielded highly accurate predictions in terms of MSE, the high MAPE values warrant further investigation. Additionally, the consistency in hyperparameters across different predictions could be indicative of a stable underlying structure in the data. It may be beneficial to explore model ensembles or more sophisticated regression techniques to address instances where the current model's performance deviates from the actual values.

References

1. NSE Historical Reports https://www.nseindia.com/resources/historical-reports-capital-market-daily-monthly-archives
2. Practical SQL – Second Edition, Debarros A., https://nostarch.com/practical-sql-2nd-edition
3. Data Engineering with Python, Crickard P., https://www.packtpub.com/en-fi/product/data-engineering-with-python-9781839214189?type=ebook
4. Python for Algorithmic Trading, Hilpisch Y., https://www.oreilly.com/library/view/python-for-algorithmic/9781492053347/
5. Order based versus level book trade reporting: An empirical analysis – Journal of Banking and Finance, James U., Thomas M., Hardy J., https://ideas.repec.org/a/eee/jbfina/v125y2021ics0378426621000327.html
6. Technical Analysis of the Financial Markets – New York Institute of Finance, Murphy J., https://www.amazon.com/Technical-Analysis-Financial-Markets-Comprehensive/dp/0735200661
7. Options, Futures, and Other Derivatives – Pearson, Hull J., Basu S, https://www.pearson.com/en-us/subject-catalog/p/options-futures-and-other-derivatives/P200000005938/9780136939917