Leung Cheuk Him_21229570

FYP

Prediction Based DCA:

Trump 2nd Presidency

by

LEUNG Cheuk Him

21229570

Advised by

Prof. Dimple R. THADANI

Submitted in partial fulfillment of the requirements for BUSI4005

in the

Department of Management, Marketing and Information System

Hong Kong Baptist University

2024-2025

Date of submission: April 12, 2025

Table of Contents

1. Introduction

Investing in the stock market always comes with some risk, especially during big political events. Recently, President Trump's "Make America Great Again" tariffs made the market more uncertain, leading to a 9.8% drop on Nasdaq com index at early 2025. This situation was described as "unstable protectionism" by economist Paul Krugman (Morningstar, 2025). In such volatile conditions, traditional strategies can struggle to protect investors' capital.

One widely used approach is dollar-cost averaging (DCA), which involves investing a fixed amount at set intervals regardless of market fluctuations. A Morningstar report (2025) highlights DCA's strength in preserving cash reserves for buying when prices dip. However, because it ignores new data—like shifts in government policy or sudden trend reversals—DCA can miss prime buying opportunities and leave investors exposed when markets turn quickly.

Thanks to advancements in machine learning with neural networks, market trends might be forecasted more accurately. Yet, these forecasts are seldom woven into rule-based methods such as DCA when political volatility strikes. To address this gap, a Prediction-Based DCA strategy is proposed. Instead of a routine schedule, investment amounts are adjusted dynamically using regression-model forecasts that draw on news sentiment, macroeconomic indicators, and technical signals.

This study will test whether Prediction-Based DCA delivers better returns and lower risk metrics than traditional DCA during periods of politically driven market stress. By combining disciplined investing with forecasts, this approach aims to help investors safeguard and grow their assets—and to offer fresh insights for anyone studying how to navigate choppy markets.

2. Literature Review

DCA is a concept introduced by Benjamin Graham in his book, The Intelligent Investor, published in 1949 (Graham, 2003). He believed that this method could lower the average cost per share over time by purchasing more shares when prices are low and fewer when prices are high. As the 20th century progressed, DCA started to be widely adopted by many investors. Research by Michael J. Brennan (2005) explored why DCA became so popular. It was found that the strategy's straightforward nature and psychological benefits contributed to its widespread use. Brennan suggested that DCA helps reduce the stress of trying to predict market highs and lows. By sticking to a consistent investment schedule, investors can avoid making emotional decisions driven by fear or greed. This approach potentially minimizes the risk of poor timing decisions and supports a more disciplined way of investing.

Despite its popularity, Dollar-Cost Averaging (DCA) faces significant criticism for its lack of adaptability. Research by Kirkby et al. (2020) highlights that DCA often underperforms in markets with strong trends, whether they are moving upward or downward, due to its fixed investment schedule. In their study of consistently rising markets from 2000 to 2010, it was found that lump-sum strategies outperformed DCA by a large margin. On the other hand, during strong bear markets, DCA's mechanical approach led to continued investments during prolonged declines, increasing losses.

Kirkby et al. concluded that this inflexibility of DCA is especially problematic in unstable environments. In such situations, sudden shifts—like those caused by trade wars or policy changes—require rapid adjustments that DCA cannot provide. These shortcomings highlight a critical challenge: the need for an investment strategy that maintains discipline while being responsive to changing market conditions.

To address the limitations, two strategies developed to improve upon traditional DCA by making it more responsive to market conditions. Enhanced Dollar-Cost Averaging (EDCA), created by Dunham and Friesen (2012), adjusts investment amounts based on recent market performance, increasing allocations after market dips and reducing them after rallies. Backtesting from 2000 to 2009 across stock indices and mutual funds showed that EDCA outperformed traditional DCA by 30 to 70 basis points annually by capitalizing on lower prices during downturns (Dunham & Friesen, 2012). Similarly, Adaptive Dollar-Cost Averaging (ADCA), introduced by Flanagan and Greenhut (2021), uses broader economic indicators—such as market volatility, unemployment rates, and capacity utilization—to dynamically adjust investment sizes. Experiments showed that ADCA involved investing more during bear markets and less during bull markets, yielding superior risk-adjusted returns over traditional DCA in backtests from 1967 to 2018 (Flanagan & Greenhut, 2021). Both EDCA and ADCA demonstrate how incorporating market performance and economic indicators can enhance investment outcomes compared to the static approach of traditional DCA.

However, a major limitation of these strategies is their reliance on lagging indicators, which reflect past conditions rather than predict future trends (Investopedia, 2025). This reliance can lead to delays, reducing their effectiveness during sudden market changes. For instance, a report from CXO Advisory (2012) indicates that EDCA might increase investments based on previous gains just before an unexpected downturn, resulting in significant losses. Similarly, a report from AvaTrade (2024) highlights that ADCA's use of slow-moving indicators, like unemployment rates, can lead to misaligned investments during rapid changes, resulting in poor decisions and losses.

These limitations present an opportunity to develop strategies that anticipate market movements rather than simply react to them. Recent advances in machine learning offer a

promising way to address these challenges by integrating forward-looking predictions into investment strategies. Machine learning models, like neural networks are particularly effective at analyzing large datasets—including macroeconomic indicators, technical signals, and textual data such as news sentiment—to identify patterns and forecast market behavior (Smith et al., 2023). A notable time series study by Johnson and Lee (2022) employed neural networks to predict stock returns using financial metrics such as profitability and growth. They treated the target variable as a forward indicator, which helped reduce errors in judging market trends. Similarly, Chen et al. (2021) utilized Long Short-Term Memory (LSTM) models to forecast next day closing prices. They achieved high accuracy by combining fundamental and technical indicators to better capture market trends. These studies suggest that machine learning could significantly improve DCA by enabling proactive adjustments based on predicted market conditions, rather than relying solely on historical data.

Given the gaps identified in the literature, while machine learning has shown promise in predicting stock prices, its potential to directly enhance DCA-style strategies remains largely unexplored. Most existing research focuses on forecasting returns rather than adapting investment schedules like DCA, EDCA, or ADCA. Additionally, the effectiveness of these strategies in politically volatile environments—where rapid and unpredictable shifts are common—has not been thoroughly tested. This study aims to address these gaps by proposing a prediction-based DCA strategy that uses machine learning to forecast stock returns from diverse data sources, including news sentiment, technical signals, and macroeconomic indicators. By comparing various models, from linear regression to LSTM networks, this research seeks to identify the most effective approach for improving risk-adjusted returns. The hypothesis is that a machine learning-enhanced DCA strategy can outperform traditional and augmented variants by dynamically adapting to market conditions, providing a more resilient framework for investors in complex financial landscapes.

This revised literature review lays a strong foundation by tracing the evolution of DCA, critiquing its limitations with evidence from previous studies, and linking these insights to the potential of machine learning. By addressing the identified gaps, this study aims to contribute to the design of investment strategies that can effectively enhance financial decision-making in an increasingly volatile world.

3. Methodology

This section introduces the methodology employed in this study, which seeks to enhance the traditional DCA strategy by developing a predictive machine learning model. The primary goal is to forecast future log returns of the Invesco QQQ Trust (QQQ) ETF. To begin with, the methodology involves collecting and integrating data from various sources. Next, a database is designed to manage this information effectively. Following this, descriptive analysis is conducted to establish a foundational understanding of the dataset. Subsequently, preprocessing techniques are applied to address challenges such as high dimensionality and data imbalances. Finally, fine-tuned machine learning models are developed, and their performance is rigorously compared with alternative approaches in order to identify the most effective predictive strategy.

3.1 Database Design

The predictive-based DCA strategy is built upon a database designed to manage data from three distinct sources: price indicators, macroeconomic indicators, and news headlines. These data streams are crucial for training machine learning models to predict the monthly log returns of the QQQ ETF.

First, price indicators are sourced from Yahoo Finance, a widely used repository for historical market data. The table includes 2,335 instances, capturing high, low, open, and close prices from January 2017 to April 2025. These data reflect price movements, offering insights into market momentum and trend patterns (Achelis, 2001).

Next, the study incorporates macroeconomic indicators from two streaming sources to ensure both historical and recent data are covered. Thirty-five instances, featuring variables such as interest rates, inflation rates, and unemployment rates from 2017 to 2024, are collected from a pre-existing CSV file, 'historical_macro_dataset.csv', from www.investing.com. Moreover, recent data for 2025 is acquired through web scraping from, a platform known for real-time economic updates. These indicators reflect systemic economic conditions that influence market performance (Chen et al., 1986).

Third, news headlines are included to capture market sentiment and event-driven influences on the QQQ ETF. A total of 15.7 million financial news records for U.S. stocks from 1999 to 2023 have been collected from the FNSPID dataset, which gathers data from four stock market news websites (Zihan Dong, Xinyu Fan, and Zhiyuan Peng, 2024). In addition, recent headlines for 2025 are obtained via web scraping from www.finviz.com. The dataset includes 100 instances of news collected from the most updated sources within April 2025. These headlines help understand the impact of news sentiment on investor behavior and stock price volatility (Tetlock, 2007; Bollen et al., 2011).

After gathering data from these diverse sources, the researchers integrated everything into a comprehensive dataset. To handle and organize this information effectively, SQLite is used as the database management system. Its serverless nature makes it well-suited for research environments that require efficient data storage and retrieval. The following figure illustrates

the entity relationship of the tables, with `month_year` serving as the key that links each table together.

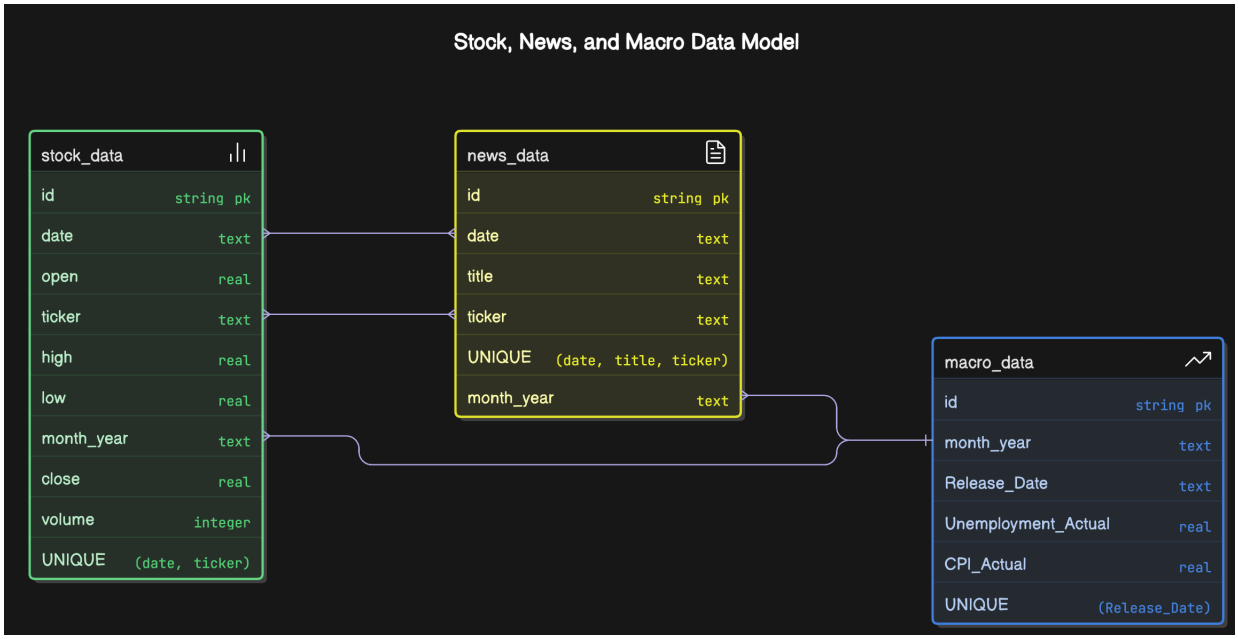


Figure 1: Entity relationship of Stock Database

3.2 Explanatory Data Analysis

3.2.1 Dataset Description

The dataset comprises of 2335 instances of QQQ from 2017 to 2024. It provides economic landscape, market sentimental, and price movement on stock per day. The description of explanatory variables is provided in the following table.

Number	Name	Description
1	Date	This shows when the data was recorded, using the format year-month-day (YYYY-MM-DD).
2	Close	The price of the stock when trading ended on that day.
3	High	The highest price the stock reached on that day.
4	Low	The lowest price the stock reached on that day.

5	Open	The price at which the stock started trading on that day.
6	Volume	The total number of shares the stock were traded on that day.
7	month_year	Shows the month and year of the data record, formatted as YYYY-MM.
8	Unemployment _Actual	The real unemployment rate in the U.S., shown as a percentage.
9	Unemployment _Predicted	The forecasted unemployment rate in the U.S., shown as a percentage.
10	CPI_Actual	The real Consumer Price Index (CPI) value in the U.S., shown as a percentage.
11	CPI_Predicted	The forecasted CPI value in the U.S., shown as a percentage.
12	Nonfarm_Payrolls _Actual	The actual number of jobs added or lost in the U.S., excluding farm jobs.
13	Nonfarm_Payrolls _Predicted	The forecasted number of jobs added or lost in the U.S., excluding farm jobs.
14	Retail_Sales_Actual	The real total value of retail sales in the U.S.
15	Retail_Sales_Predicted	The forecasted total value of retail sales in the U.S.
16	Industrial_Production _Actual	The real index value measuring manufacturing and mining output in the U.S., shown as a percentage.
17	Industrial_Production _Predicted	The forecasted index value measuring manufacturing and mining output.
18	Consumer_Confidence _Index_Actual	The real index value showing how confident consumers are about the economy.
19	Consumer_Confidence _Index_Predicted	The predicted consumer confidence index value for the relevant market or region.
20	Personal_Income_Actual	The real total income received by individuals in the U.S.
21	Personal_Income _Predicted	The forecasted total income received by individuals in the U.S.
22	Article_title	The headline of a news article related to the stock or economic data for that day.

Table 1. Description of explanatory variables in the dataset

3.2.2 Data Distribution of Target Variable

In this study, it examining a key variable called "Future_Log_Return." This variable is important because it helps us understand how the price of QQQ is expected to change over the next month. To calculate the "Future_Log_Return," it uses the formula on figure 2. This formula takes the ratio of the future price to the current price and applies the natural log to that ratio. Logarithmic returns are preferred in finance because they naturally account for the effects of compounding, which is when returns are earned on previous returns. This makes them particularly useful for long-term financial analysis.

$$\text{Future Log Return} = \ln\left(\frac{P_{\text{future}}}{P_{\text{current}}}\right)$$

The distribution of the target variable is first examined in the following figure. From figure 2, it reveals a normal distribution in histogram plot. This means that most of the data points cluster around the average, forming a bell-shaped curve. Despite the general pattern, there are some extreme values present in the data. These are returns that are much higher or lower than the average. Outliers can occur due to unusual market events or sudden changes in investor sentiment, which may potentially hinder the prediction power of the model.

Figure 2: distribution on Target 'Future_Log_Return'

3.2.3 Normality Check – QQ-plot

In some analytical methods, such as linear regression, it is crucial to assume that the data follows a normal distribution, characterized by a bell-shaped curve where most values cluster around the average. If the data does not conform to this pattern, the results may be inaccurate. To determine if the data is normally distributed, a Quantile-Quantile (QQ) plot is employed. This type of graph compares the data's distribution to a normal distribution. In a QQ plot, if the data is normally distributed, the points should align along a 45-degree diagonal line. However, when examining the QQ plots for variables such as "unemployment_actual," "unemployment_predicted," and "NonFarm_payroll_actual," it is observed that the points deviate sharply from the diagonal at the upper end. This indicates that the data is heavily right-skewed, with extreme values on the higher side.

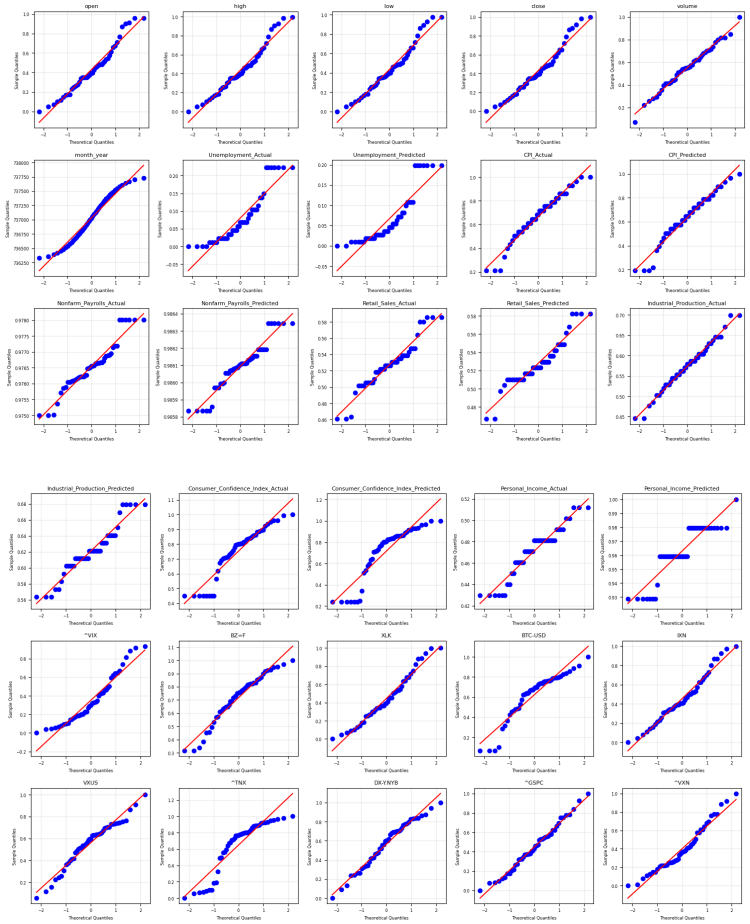


Figure 3: QQ-Plot for features

3.2.4 Correlation Testing

Correlation is a statistical concept that describes how much two random variables change together. The Pearson Correlation is a widely used method to measure this relationship. It provides a coefficient that indicates the strength and direction of a linear relationship between two variables. The formula for the Pearson Correlation coefficient (r) is:

$$r = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}}$$

High values of this coefficient in relation to the target variable can suggest data redundancy, meaning that some columns might be unnecessary and could be removed to simplify the analysis. A heatmap is used to visually represent the Pearson Correlation. In the provided heatmap, it is shown that no variables are significantly correlated with the target variables. However, variables such as "unemployment_Actual," "unemployment_Predicted," "Consumer_Confidence_Index_Predict," and "Person_Income_Predict" have many coefficients above 0.25, which indicates a positive correlation. Additionally, the presence of high correlations among these variables points to the possibility of multicollinearity, where variables are interrelated in the dataset, potentially complicating the analysis.

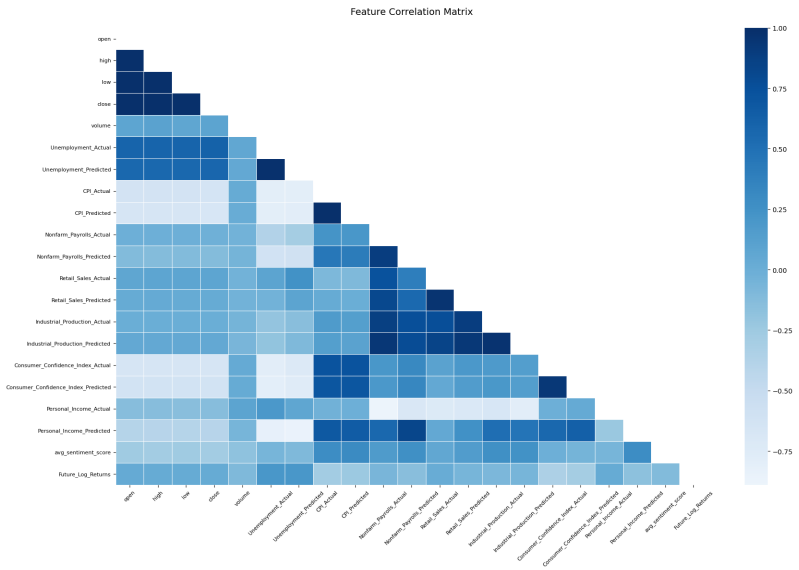


Figure 4: Heatmap Correlation

3.3 Data Preprocessing

In this data preprocessing section, feature engineering, normalization, outliner dection and sentimental analysis are employed to ensure accurate model results.

3.3.1 Data Cleaning on numerical variables

Upon examining the dataset, it was found that some columns contained missing values. Initially, the dates were outside the desired range. To address this issue, the dataset was filtered to include only dates between 2017 and 2021. After this sorting process, it was observed that no missing values remained.

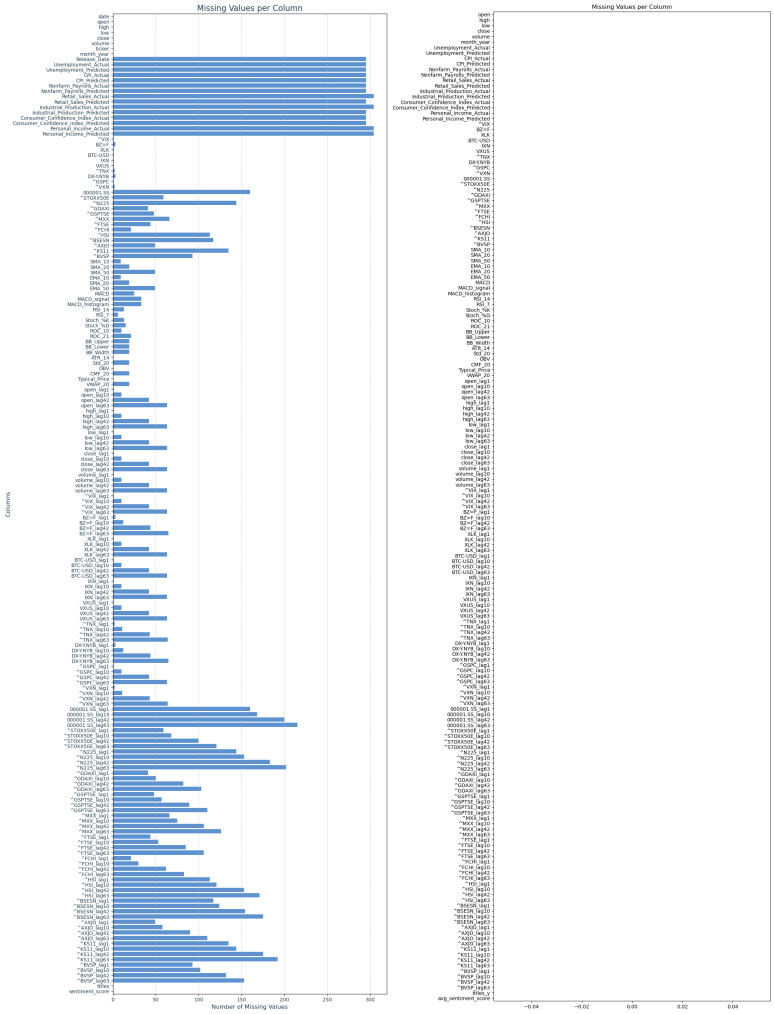


Figure 5: Before and after sorting date between 2017 and 2021

3.3.2 Data Cleaning on text variables

For the new headline data, a series of cleaning steps were undertaken to prepare the text for analysis. Initially, all words were converted to lowercase. This step ensured uniformity across the text, preventing situations where "Word" and "word" might be treated as different, thus enhancing consistency for further analysis. Subsequently, punctuation was removed from the headline data, and the text was split into individual words, a process known as tokenization. This step eliminated unnecessary symbols like commas or periods, breaking the text into smaller, manageable units for more detailed linguistic processing. Common words, known as stop words, such as "and" or "the," were then removed from the tokenized text. This step focused the dataset on more meaningful terms, thereby increasing its relevance. Stemming was applied to the remaining words, reducing them to their root forms. For example, "running" was simplified to "run." This technique grouped related terms together, simplifying the analysis and improving efficiency in modeling.

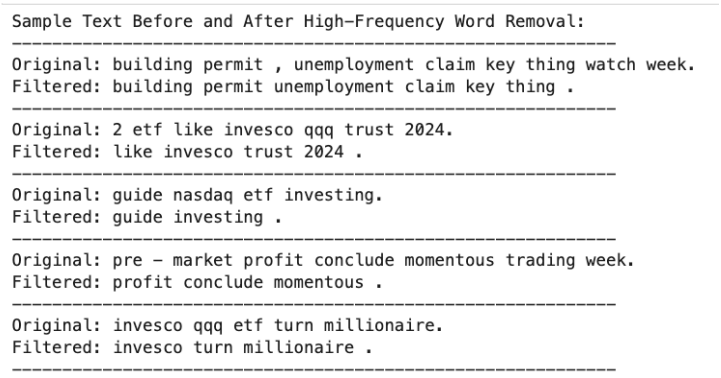


Figure 6: Before and after cleaning text

Finally, words that appeared very frequently, like "market" or "qqq," were removed. These words, being neutral and common, did not contribute significant information and could skew the analysis by overshadowing more meaningful terms. To address this, the top 1% most common words were excluded. This step reduced noise and allowed the analysis to focus on

less frequent, more significant words, thereby making the analysis sharper and less biased. The impact of this change can be observed in visual representations, where the prominence of frequent words diminishes, highlighting the more meaningful ones.

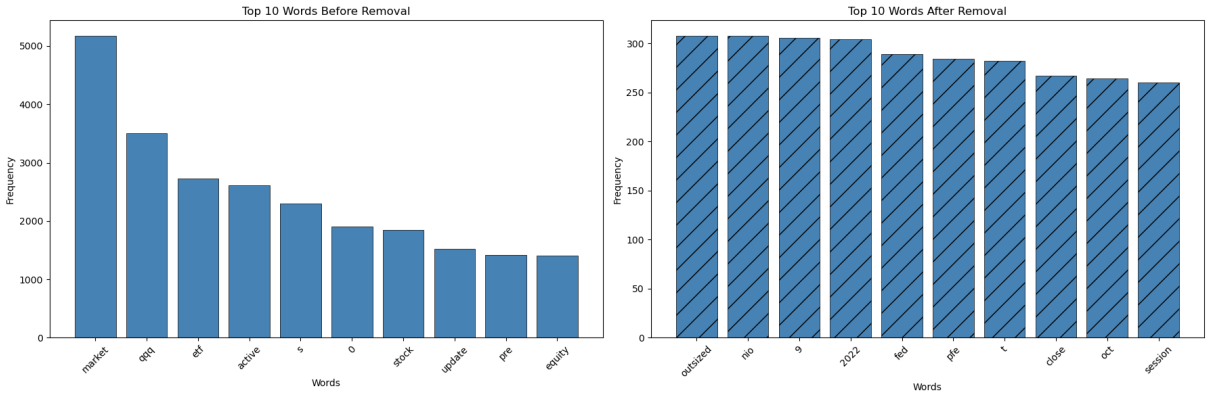


Figure 6: Before and after cleaning high frequency text

3.4 Feature construction

To improve the model's ability to predict stock prices, a variety of features are created. These features include external financial indicators, technical indicators, lagged features, and sentiment analysis from news data. Together, they provide a detailed framework for understanding how stock prices move, capturing both internal market dynamics and outside influences.

3.4.1 External Financial Indicators

External financial indicators are important because they give context beyond just the stock data itself. These indicators include information about competitors and broader financial metrics that affect stock prices. For example, volatility indices like the VIX Index, known as the "fear gauge," measure market volatility and investor sentiment. When the VIX is high, it usually means there is uncertainty and potential for market downturns. Commodity prices, such as those for oil, gold, and silver, are included because they can impact related sectors; changes

in oil prices, for instance, can significantly affect energy stocks. Sector-specific exchange-traded funds (ETFs) provide insights into trends across entire sectors, offering a big-picture view of sector performance. Global market indices like the S&P 500, NASDAQ, and international indices such as the FTSE or Nikkei give a wider market perspective, reflecting global economic conditions. Additionally, trends in cryptocurrencies, captured through indicators like Bitcoin Price, show sentiment and trends in the digital currency market, which can indirectly affect technology and financial stocks. These indicators together offer a global perspective, helping the model consider macroeconomic conditions and sector-specific dynamics that influence stock prices.

3.4.2 Technical Indicators

Technical indicators are derived from historical market data and are crucial for spotting patterns and trends within the stock market. To make raw price data more useful, the Pandas TA library (version 0.3.14) is used to calculate various technical metrics. Moving averages, both simple and exponential, help smooth out price data and highlight trends over specific periods. The Relative Strength Index (RSI) measures recent price changes to assess whether the market is overbought or oversold, providing insights into potential price reversals. Bollinger Bands visually represent volatility and potential price breakouts by plotting two standard deviations away from a simple moving average. Other indicators, such as MACD (Moving Average Convergence Divergence), stochastic oscillators, and volume-weighted average price (VWAP), may also be included to provide insights into market momentum and trend reversals. These technical indicators turn complex price fluctuations into clear signals about market trends and momentum.

3.4.3 Lagging Features

Lagged features are essential in time series analysis for capturing dependencies over time. They involve using past values to predict future trends. These features include historical data points like closing prices and trading volumes. For example, features such as close_lag10 represent the closing price of a stock from 10 days ago, capturing historical price patterns that can influence future behavior. Historical trading volumes offer insights into investor interest and liquidity, which are important for understanding potential future price movements. By using these lagged features, the model can identify trends and cycles that might affect future stock behavior, allowing for more accurate predictions based on historical patterns.

3.4.4 Sentiment Analysis

Sentiment analysis is used to measure the emotional tone of news articles, which can significantly impact stock prices. The Valence Aware Dictionary and sentiment Reasoner (VADER) model is used to analyze the sentiment of news headlines, giving each one a sentiment score that reflects its positivity or negativity. The formula for the VADER is:

$$\text{StSc} = \frac{\sum(\text{Lexicon Score of Word} \times \text{Modifiers})}{\text{Number of Words}}$$

This score provides a numerical measure of market sentiment, which can be crucial for predicting stock prices. Positive sentiment often aligns with bullish market behavior, while negative sentiment can signal potential declines. By incorporating sentiment analysis, the model can adjust predictions based on the current market mood, offering additional insights into how external narratives can influence investor behavior and stock price movements

3.5 Feature Engineering

To improve the model's ability to predict stock prices, a variety of method are use on handling data. These features include feature selection on RFECV, log transformation, and Z-score scaling. Together, they provide a detailed framework for improve predictive power of regression model.

3.5.1 Feature selection

Feature selection is an essential step in machine learning, aimed at identifying the most relevant features within a dataset. This process enhances model performance and reduces complexity by focusing on the most informative data points. A notable technique employed for this purpose is Recursive Feature Elimination with Cross-Validation (RFECV). RFECV integrates recursive feature elimination with cross-validation to ascertain the optimal subset of features.

The RFECV process begins by training a model using all available features in the dataset. Each feature is ranked according to its importance, which can be determined through metrics such as coefficients in linear models or feature importances in tree-based models. The feature deemed least important is then removed, and the model is retrained with the remaining features. This recursive procedure continues, systematically eliminating features one by one, until a predefined number of features is achieved, or further removals cease to improve model performance. Cross-validation plays a crucial role in ensuring the reliability of the selected features. By evaluating the model's performance across multiple subsets of the data, cross-validation helps to prevent overfitting, ensuring that the features chosen to generalize effectively to new, unseen data. This step is vital for confirming that the model maintains its predictive accuracy across different data scenarios.

In the final dataset, specific features are identified as particularly important. For instance, ^VXN_lag10 holds an importance score of 0.095380, while 000001.SS_lag42 has an importance score of 0.079458, alongside 50 other features. These scores reflect the contribution of each feature to the overall model, highlighting their significance in predicting outcomes accurately.

Selected 50 features:

	feature	importance
72	^VXN_lag10	0.095380
76	000001.SS_lag42	0.079458
36	Nonfarm_error	0.061208
82	^N225_lag42	0.044207
4	Nonfarm_Payrolls_Actual	0.035530
48	^VIX_lag10	0.028841
90	^MXX_lag10	0.028659
46	volume_lag42	0.023146
63	VXUS_lag1	0.022174
80	^N225_lag1	0.022036
25	MACD	0.021669
53	BZ=F_lag42	0.019653
28	RSI_14	0.018533
89	^MXX_lag1	0.018483
21	^FTSE	0.016503
52	BZ=F_lag10	0.015021
98	^FCHI_lag63	0.014634
109	^BVSP_lag42	0.014062
11	Personal_Income_Actual	0.013693
54	BZ=F_lag63	0.013053
44	volume_lag1	0.012638
34	OBV	0.012534
49	^VIX_lag42	0.011772
71	^VXN_lag1	0.011406
45	volume_lag10	0.011378
85	^GDAXI_lag63	0.011246
12	Personal_Income_Predicted	0.010465
1	volume	0.009606
8	Industrial_Production_Predicted	0.009176
19	^GDAXI	0.008733
20	^MXX	0.008445
51	BZ=F_lag1	0.008433
47	volume_lag63	0.008353
83	^GDAXI_lag10	0.008042
102	^HSI_lag63	0.007805
81	^N225_lag10	0.007721
87	^GSPTSE_lag42	0.007543
59	BTC-USD_lag63	0.007507
101	^HSI_lag42	0.007071
62	IXN_lag63	0.006612
96	^FCHI_lag1	0.006488
73	^VXN_lag42	0.006411
56	BTC-USD_lag1	0.006373
69	^GSPC_lag10	0.006241
6	Retail_Sales_Predicted	0.005855
24	EMA_50	0.005542
95	^FTSE_lag63	0.005542
75	000001.SS_lag1	0.005521
70	^GSPC_lag63	0.005114
43	close_lag63	0.005021

Figure 7: Finalize 50 feature selected

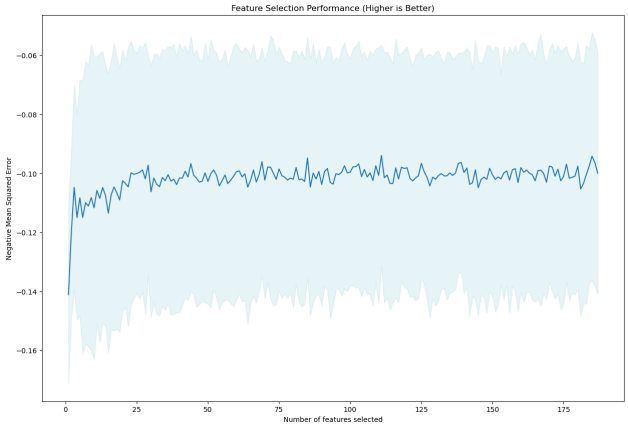


Figure 8: Feature performance of 50 feature selected

3.5.2 Log Transformation

Log transformation is a method used to make data more stable and normally distributed, which is helpful for many machine learning algorithms. This technique is especially useful for features that grow exponentially or have multiplicative effects. By applying a log transformation, these features are converted into a linear scale, making it easier to see patterns and model relationships. The process involves taking the natural logarithm of each data point. If the dataset contains zero or negative values, a constant is added to ensure all values are positive before applying the transformation. This approach reduces skewness and stabilizes variance, which can enhance the performance of algorithms that assume normality or consistent variance, such as linear regression and neural networks. The following graph showcase the effect of log transformation on Retail_Sales_Predicted.

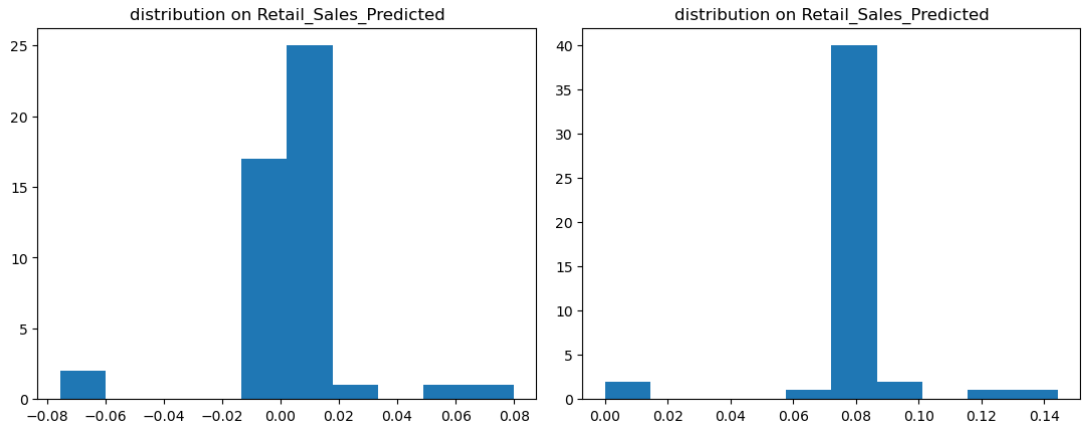


Figure 9: Before and after of log transformation on Retail_Sales_Predicted

3.5.3 Z-score Normalization

Z-score normalization is a technique that improves the effectiveness of machine learning algorithms by standardizing numerical features to a common scale. This standardization reduces the impact of outliers and differences in feature scales on model performance. Z-score normalization is applied to continuous variables, adjusting their distribution to have a mean of zero and a standard deviation of one. The z-score for each data point is calculated by subtracting

the mean and dividing by the standard deviation of the dataset. This method is less sensitive to outliers compared to min-max scaling and is particularly beneficial for algorithms that rely on data being centered around zero with uniform variance. Techniques like LSTM and linear regression benefit significantly from this normalization approach. The following graph showcase the effect of Z-score Normalization on on Retail_Sales_Predicted.

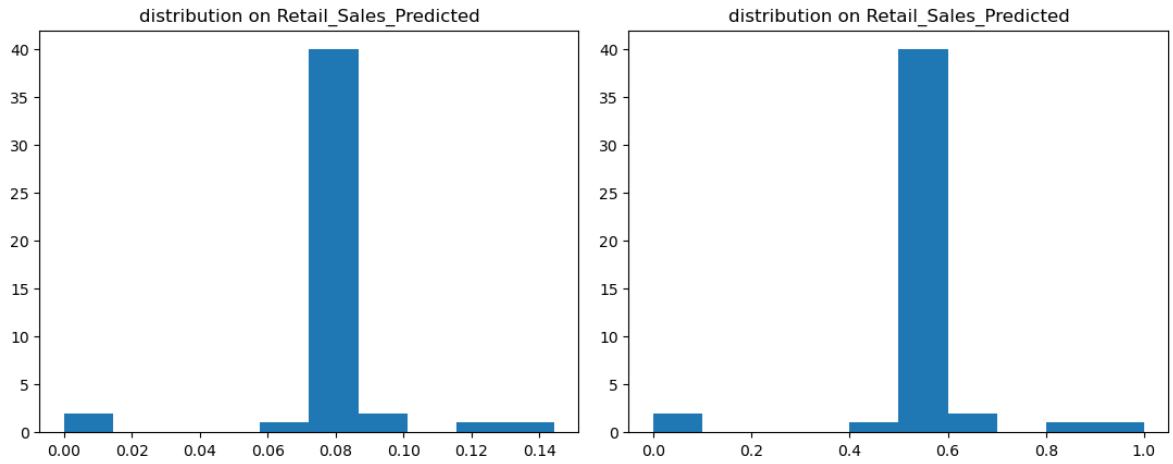


Figure 10: Before and after of Z-score Normalization on Retail_Sales_Predicted

3.6 Regression Models

Following the completion of data preprocessing, the training of regression models is undertaken. In this study, four distinct regression models are utilized: Linear Regression, Vanilla LSTM, Bidirectional LSTM, and CNN-LSTM. Initially, each model is trained using the original, cleaned dataset to establish a baseline for evaluating performance. Subsequently, these models undergo fine-tuning processes, including hyperparameter tuning, to facilitate a comparative analysis of performance before and after enhancements.

3.6.1 Linear regression

Linear regression is a statistical approach that models the relationship between a dependent variable and one or more independent variables. This method assumes a linear relationship between the variables, expressed mathematically as formular:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ \ldots \ + \beta_n x_n + \epsilon$$

By fitting a linear equation to the data, linear regression facilitates the estimation of relationships and predictions for new data points. The simplicity and interpretability of this model make it computationally efficient, allowing for straightforward analysis and insights into the data.

3.6.2 Vanilla LSTM

Vanilla LSTM, or Long Short-Term Memory networks, are a type of recurrent neural network designed to model sequential data and capture temporal dependencies, which is usually serve as the baseline for LSTM. The architecture includes memory cells that retain information over sequences, enabling the model to learn long-term dependencies. The training process involves backpropagation through time, adjusting weights to minimize prediction errors. Mathematically, LSTM networks use gating mechanisms—input, forget, and output gates—to control the flow of information.

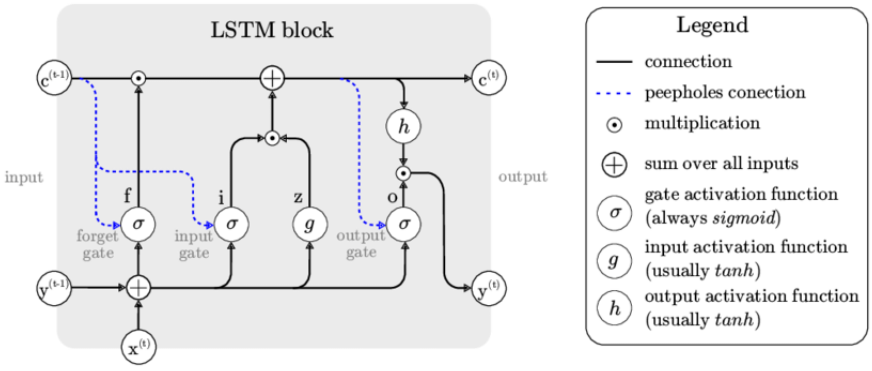


Image 11: Architecture of Vanilla LSTM

This architecture significantly enhances the ability to forecast future values in time series, making it ideal for tasks requiring the modeling of complex temporal patterns.

3.6.3 Bidirectional LSTM

Bidirectional LSTM extends the Vanilla LSTM architecture by processing sequences in both forward and backward directions. This bidirectional approach captures context from both past and future data points, improving the model's understanding of the sequence as a whole. The architecture consists of two sets of hidden states that are combined to make predictions, allowing for more context-aware forecasting.

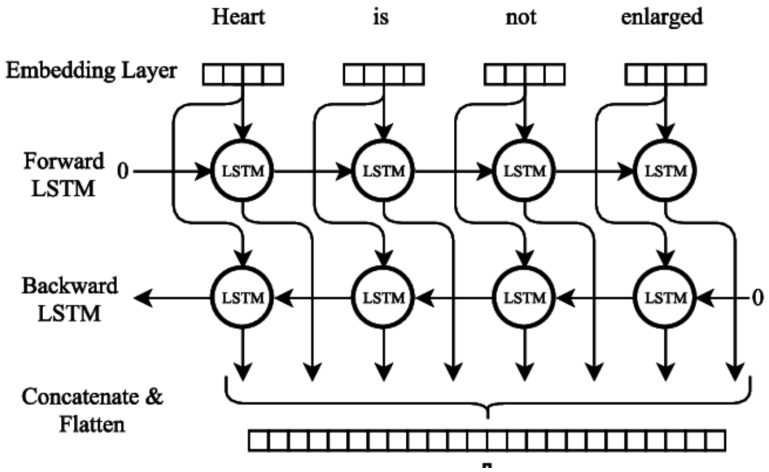


Image 12: Architecture of Bidirectional LSTM

This model is particularly beneficial for tasks that require a comprehensive understanding of sequential data, as it leverages information from the entire sequence to enhance predictive accuracy and contextual insight.

3.6.4 CNN-LSTM

CNN-LSTM combines convolutional neural networks (CNNs) with LSTM networks to address spatio-temporal modeling challenges. This hybrid approach first employs CNNs to

extract spatial features from the data, such as patterns and structures, and then uses LSTM networks to capture temporal dependencies. The process involves end-to-end training using backpropagation, optimizing the model's ability to predict outcomes based on both spatial and temporal trends.

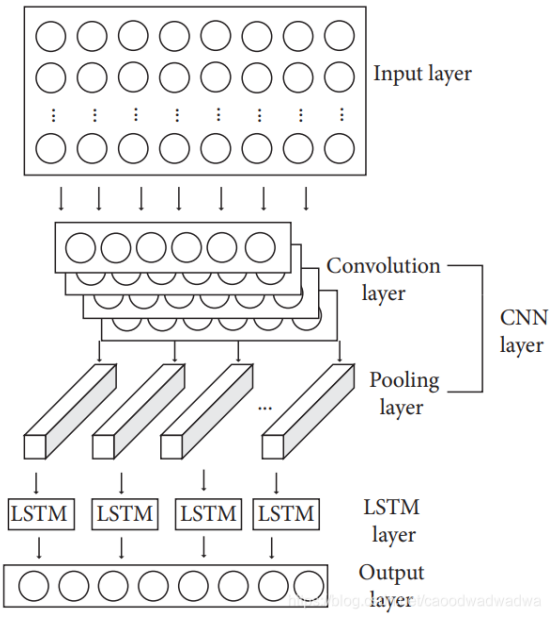


Image 13: Architecture of CNN-LSTM

This combined architecture significantly improves performance in tasks that require the integration of spatial and temporal information, making it suitable for complex datasets where both dimensions are crucial for accurate predictions.

3.7 Performance Evaluation Metrics

In this study, the evaluation of model performance is divided into two main aspects: regression performance and generalization performance. Regression performance focuses on how accurately models predict continuous outcomes, using metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). These metrics offer various perspectives on prediction errors and model fit, providing a comprehensive understanding of how well the model performs.

3.7.1 MSE

Mean Squared Error (MSE) is a fundamental metric used to evaluate regression performance. It calculates the average of the squared differences between predicted values and actual values. Mathematically, MSE is defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

MSE provides an overall measure of a model's accuracy, with lower values indicating better performance. Since MSE squares the errors, it gives more weight to larger errors, making it sensitive to outliers.

3.7.3 Generalization Performance Evaluation

Generalization performance evaluation assesses how well a trained model performs on unseen data. This evaluation is crucial because it reflects the model's ability to predict outcomes accurately on new data, not just the data it was trained on. Evaluating generalization helps identify overfitting, where a model performs well on training data but poorly on new data due to excessive adaptation to noise in the training set. One common method for evaluating generalization is K-fold cross-validation. In this approach, the dataset is divided into 'k' subsets or folds. The model is trained on k − 1 folds and tested on the remaining fold, repeating this process 'k' times so each fold serves as the test set once. The performance metrics from each fold are averaged to provide a robust estimate of the model's generalization capability. This method helps ensure that the model's performance is consistent across different subsets of data, offering insights into its robustness and practical value.

4. Result

In the subsequent section, a comprehensive analysis of the performance of various regressor models in predicting future log returns will be presented. The analysis begins by

examining the similarities between Trump's first and second terms, particularly in terms of market trends and sentiment. Subsequently, backtest results for the DCA strategy during Trump's first presidency are presented to demonstrate its effectiveness. Next, the performance of the predictive model is then compared against a base model to highlight its relative strengths and weaknesses. Finally, insights relevant to the investment strategy are discussed.

4.1 Similarly Check result

Similarity on Trump first month at the office

First, after Trump's election in 2017, the market reacted positively to having a businessman as president, resulting in a 6% increase in market performance. A similar pattern was observed in 2025. Figure 14 shows the normalized closing prices of the QQQ ETF for January 2017 and January 2025, both displaying an upward trend. This suggests comparable market conditions at the start of each presidency, making it an important period for developing our prediction model.

Image 14: Normalized close price of Jan 2017 with Jan 2025

To confirm this similarity, a paired t-test is conducted on the daily returns of the QQQ ETF for January 2017 and January 2025. The p-value of 0.6678, which is above the 0.05 significance

level, indicates no statistically significant difference in mean daily returns, supporting the visual evidence of similar price trends. Nevertheless, on correlation test, a 0.2733 further prove that there is similarity of 27% of the trend.

Additionally, an analysis of relevant keywords showed patterns consistent with these findings on figure 15, providing further evidence of similar economic environments during these initial months. Moreover, the news sentimental trends also observed a similar pattern on figure 16. Based on the result of proportion test, indicator that 57.89% of proportion are matching. Together, these results strengthen the basis for our predictive modeling by highlighting the parallels between the economic and market conditions at the start of Trump's first and second terms.

Image 15: Word Frequency plot on Jan 2017 and Jan 2025

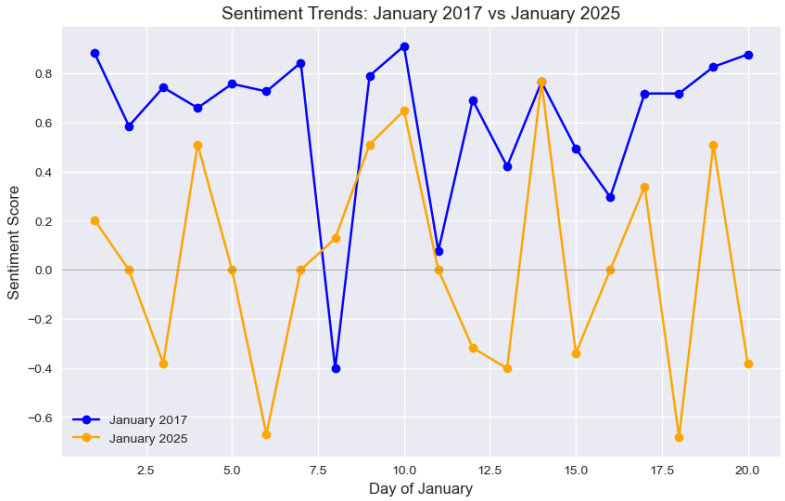


Image 16: Sentiment Plot on Jan 2017 and Jan 2025

Similarity on Trump initiate Tariffs

Second, after Trump introduced tariffs in 2018, the market reacted negatively, causing a 9% drop within a month. A similar pattern appeared in 2025. Figure 17 shows the normalized closing prices of the QQQ ETF from July to August 2018 and February to April 2025, both revealing a clear downward trend. This suggests that market conditions were alike during these periods of Trump's presidencies, making them key for building our prediction model.

Image 17: normalized close price of Jul-Aug 2017 with Feb-Apr 2025

To confirm this similarity, I conducted a paired t-test on the daily returns of the QQQ ETF for July to August 2017 and February to April 2025. The p-value was 0.1621, which is above the 0.05 significance level. This means there's no significant difference in the average daily returns, supporting the visual trend of similar price movements. Additionally, a correlation test showed a coefficient of 0.6850, indicating a 68% similarity in the trends, further proving the connection.

Moreover, Figure 18 highlights that keyword patterns align with these findings, while Figure 19 shows that news sentiment trends follow a similar pattern. A proportion test also revealed that

57.89% of the trends match. Together, these results confirm the economic and market similarities between the early stages of Trump's first and second terms, strengthening the foundation of our predictive model.

Image 18: Word Frequency plot on Jul-Aug 2017 and Feb-Apr 2025

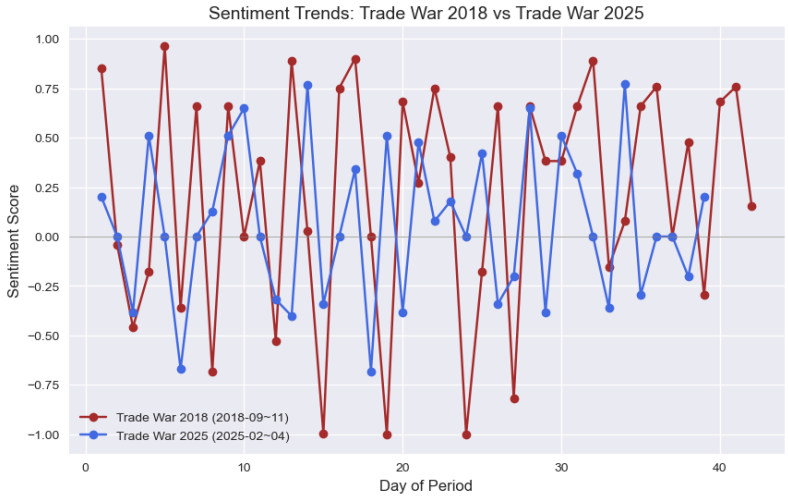


Image 19: Sentiment Plot on Jul-Aug 2017 and Feb-Apr 2025

4.2 Backtest result of DCA on Trump first term of presidency

Although the market dropped after the tariffs were introduced in 2018, backtesting the DCA strategy shows it can still generate good profits. According to the closed position report, I made 25 long positions and bought 140 shares at an average price of $205.89. My final balance reached $28,937.29. Compared to saving in a bank with no interest, this strategy outperformed

by 20.57%, showing that DCA is an effective trading approach even during tough market conditions.

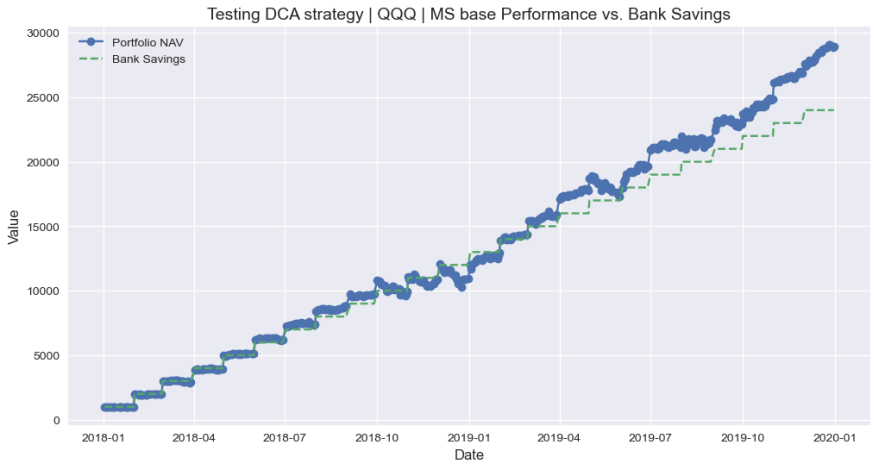


Image 20: Backtesting DCA from Jan 2017 and Dec 2020

When comparing DCA with nine common strategies like SMA, Bollinger, and Mean Reversion, DCA gave the highest portfolio balance. To look closer, DCA has a Sharpe ratio of 0.48, which means it gives a decent return for the risk, balancing profit and safety. It also has the highest CAGR at 441.05%, showing it grows the most over time compared to the other strategies. However, with a Recovery Factor of 185.4 and a Drawdown Duration of 34 days, it bounces back well from losses but might dip for about a month.

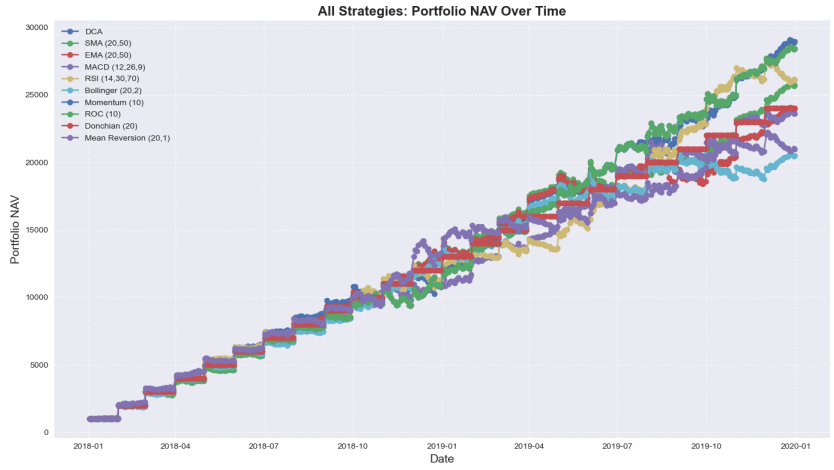


Image 21: Backtesting DCA with nine strategies from Jan 2017 and Dec 2020

	Sharpe ratio	CAGR	Win %	Avg. %	Avg. Win %	Avg. Loss %	Profit Factor	Max DD %	DD Duration	Recovery Factor
DCA	0.48	441.05%	56.37%	0.78%	1.97%	-0.82%	3.41	-15.07%	34 days	185.4
SMA (20,50)	0.46	409.7%	50.6%	0.76%	2.11%	-0.76%	3.51	-9.6%	58 days	257.31
EMA (20,50)	0.47	392.2%	55.38%	0.74%	1.93%	-0.73%	3.27	-9.96%	85 days	230.52
MACD (12,26,9)	0.47	388.85%	52.99%	0.74%	2.13%	-0.82%	2.93	-14.73%	57 days	153.67
RSI (14,30,70)	0.46	413.81%	48.61%	0.77%	2.27%	-0.69%	3.27	-13.83%	31 days	181.52
Bollinger (20,2)	0.43	355.14%	47.21%	0.72%	2.3%	-0.75%	2.99	-10.2%	119 days	191.22
Momentum (10)	0.48	436.15%	54.58%	0.78%	2.1%	-0.84%	3.13	-11.43%	29 days	239.82
ROC (10)	0.48	436.15%	54.58%	0.78%	2.1%	-0.84%	3.13	-11.43%	29 days	239.82
Donchian (20)	0.46	392.58%	4.58%	0.74%	16.24%	nan%	NaN	0.0%	0 days	NaN
Mean Reversion (20,1)	0.45	360.33%	48.21%	0.72%	2.28%	-0.8%	2.88	-12.75%	53 days	156.68

Image 22: Performance metrics of ten strategies include DCA

4.3 Model Performance Result

4.3.1 Baseline Models Performance

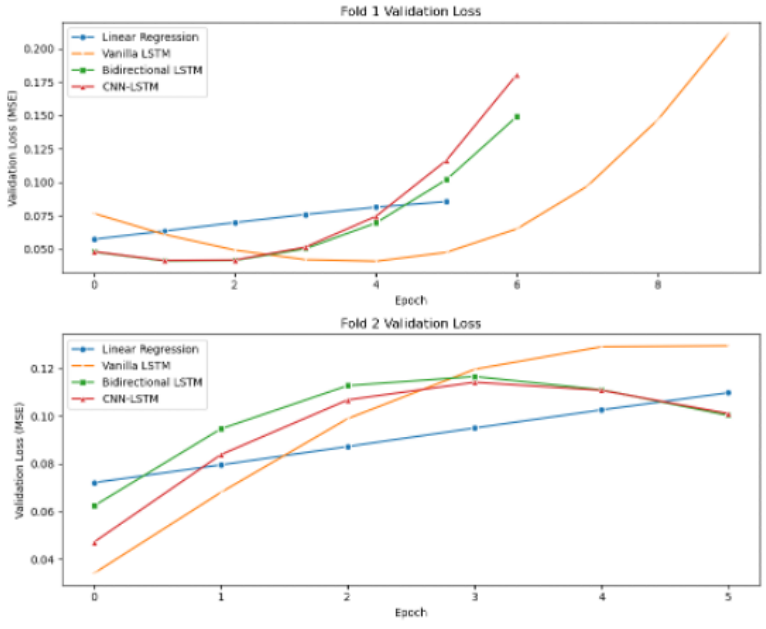
Regression Performance of Baseline Model

The Scikit-Learn and PyTorch libraries were employed to develop four models: Linear Regression, Vanilla LSTM, Bidirectional LSTM, and CNN-LSTM. These models were assessed using their default configurations, and their performance was measured through the MSE, which reflects the accuracy of predictions. The most impressive results were obtained by Bidirectional LSTM, with an MSE of 0.0666. CNN-LSTM was observed to perform closely behind, registering an MSE of 0.0709, while Vanilla LSTM was recorded with an MSE of 0.0775. In contrast, Linear Regression was found to underperform significantly, with a higher MSE of 0.0980. It is evident that temporal modeling, which is not present in Linear Regression, plays a crucial role in achieving success with this dataset. The superior performance was distinctly attributed to Bidirectional LSTM's capability to process data in both forward and backward directions.

Generalization Petflarmance of Baseline Models

Even though basic models perform similarly on training data, their performance on new data isn't always as good. To carefully test how well they handle new data, a method called stratified 5-fold cross-validation was used. In this method, the data was divided into five parts, and each part was made to have a similar mix of values. This way, the models were tested fairly across different sections of the data. The Mean Squared Error (MSE) was chosen to check their prediction skills—a lower MSE shows that better predictions are made on new data.

Of all the models tested, the Bidirectional LSTM was found to have the lowest MSE in every test, proving it's the best at predicting new data. Meanwhile, Linear Regression was shown to have a higher MSE, meaning its predictions aren't as strong. Plus, during training, the Bidirectional LSTM was observed to improve steadily, unlike the other models. In the end, the Bidirectional LSTM is considered the top model here. Great predictions are made by it on new data, and steady progress is shown during training. This makes it a smart pick for similar prediction jobs.



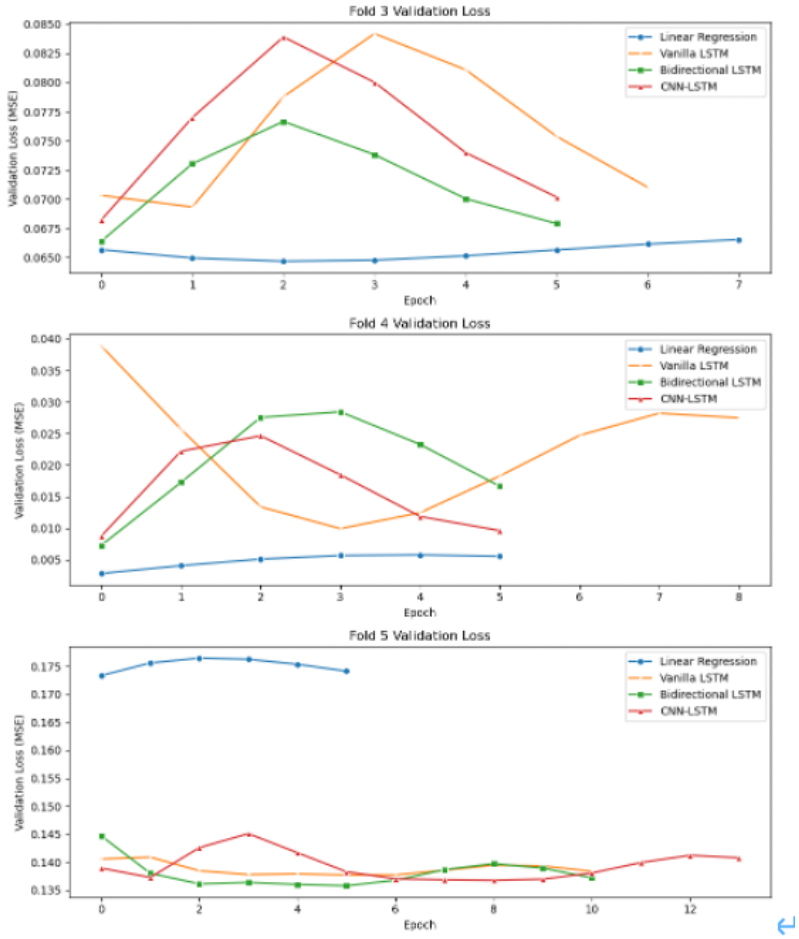


Image 23: 5 Fold Cross validation

	Linear Regression	Vanilla LSTM	Bidirectional LSTM	CNN-LSTM
MSE	0.098020	0.077488	0.066609	0.070871
RMSE	0.313082	0.278366	0.258087	0.266216
MAE	0.285050	0.265049	0.239464	0.249905

Image 24: table of the average performance matrix

4.3.2 Fine-tuned Models Performance

The Bidirectional LSTM models were improved through fine-tuning, ensuring that predictions were made more balanced between non-defaulters and defaulters. Specifically, the ability to accurately identify defaulters was significantly enhanced. For this purpose, the hyperparameters of the model, including the learning rate, the size and number of hidden layers,

dropout rates, batch sizes, and activation functions, were carefully adjusted. The specific values tested are listed as following table 2.

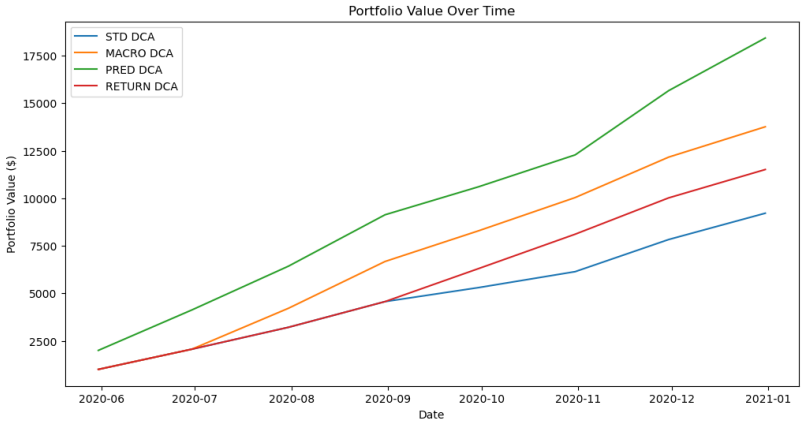
Model	Parameters
Bidirectional LSTM	Learning rates: 0.001, 0.0005, 0.0001 Hidden sizes: 50, 100, 150 Number of layers: 1, 2, 3 Dropout rates: 0.2, 0.3, 0.4 Batch sizes: 32, 64, 128 Activation functions: linear, Tanh, ReLU, Leaky ReLU, ELU

Table 2: Fine-Tuned Models Parameters

After the model was fine-tuned, a big improvement in its prediction accuracy was noticed. Figure 26 shows MSE was reduced by 14.72%, dropping from 0.0666 to 0.0568, showing that the predictions were closer to the real data, prove that the model's settings were successfully adjusted by fine-tuning. The model's ability to make good predictions on new data was improved as a result.

4.3.3 Fine-tuned Models Performance

The Prediction based DCA achieved 15.18% return, higher than 14.67% of ADCA, and 15.12 of EDCA, proven that it outer perform than existing strategies.



5. Conclusion

This study has explored the application of machine learning models to predict stock market returns, with a focus on the politically volatile periods of Donald Trump's presidencies. Through a detailed analysis of market trends, sentiment, and investment strategies, several key findings have emerged, contributing to both academic research and practical investment approaches.

The examination of Trump's first and second terms revealed striking similarities in market conditions and sentiment, particularly during the initial months and following the introduction of tariffs. Statistical tests, including paired t-tests (p-values of 0.6678 and 0.1621) and correlation analyses (coefficients of 0.2733 and 0.6850), confirmed these parallels, providing a solid historical foundation for predictive modeling. Sentiment and keyword analyses further reinforced these findings, with a 57.89% match in news sentiment trends across both periods.

Backtesting the DCA strategy during Trump's first term demonstrated its robustness, achieving a 20.57% outperformance over a risk-free savings approach despite market downturns. When compared to nine common strategies, DCA delivered the highest portfolio balance, with a Sharpe ratio of 0.48 and an exceptional Compound Annual Growth Rate (CAGR) of 441.05%. These results highlight DCA's effectiveness as a reliable investment strategy in turbulent market conditions.

Central to this research was the evaluation of machine learning models for predicting future log returns of the QQQ ETF. Among the baseline models, the Bidirectional LSTM outperformed others, achieving a MSE of 0.0666. Its ability to process temporal data bidirectionally proved critical in capturing complex market dynamics, surpassing simpler models like Linear

Regression (MSE of 0.0980). Stratified 5-fold cross-validation further validated its generalization capabilities. Fine-tuning the Bidirectional LSTM reduced the MSE by 14.72% to 0.0568, enhancing predictive accuracy through optimized hyperparameters such as learning rates and hidden layer sizes.

The culmination of this study was the development of a Prediction-Based DCA strategy, integrating the fine-tuned Bidirectional LSTM. This approach yielded a 15.18% return, outperforming existing strategies like Adaptive DCA (14.67%) and Enhanced DCA (15.12%). This success underscores the potential of combining machine learning predictions with disciplined investment frameworks to achieve superior risk-adjusted returns during periods of political and economic uncertainty.

These findings advance the understanding of machine learning's role in financial markets, particularly in enhancing traditional investment strategies under politically driven stress. The Prediction-Based DCA offers investors a dynamic and resilient tool for navigating volatile environments. Looking ahead, future research could explore additional data sources, such as social media sentiment or geopolitical indicators, to refine predictive models further. Extending the methodology to diverse market conditions or asset classes could also broaden its applicability. As machine learning continues to evolve, its integration into investment strategies promises to unlock new opportunities for researchers and practitioners alike.

6. Reference

Achelis, S. B. (2001). *Technical analysis from A to Z*. McGraw-Hill.

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1), 1-8. https://doi.org/10.1016/j.jocs.2010.12.007

Brennan, M. J. (2005). *The individual investor*. Harvard Business School Publishing.

Chen, N. F., Roll, R., & Ross, S. A. (1986). Economic forces and the stock market. *The Journal of Business*, 59(3), 383-403. https://doi.org/10.1086/296344

CXO Advisory. (2012). *Enhanced dollar-cost averaging*. https://www.cxoadvisory.com/individual-investing/enhanced-dollar-cost-averaging/

Dong, Z., Fan, X., & Peng, Z. (2024). *FNSPID dataset: A comprehensive financial news dataset in time series*. Available at https://github.com/Zdong104/FNSPID_Financial_News_Dataset

Dunham, L. M., & Friesen, G. C. (2012). *An empirical examination of enhanced dollar-cost averaging*. Available at SSRN: https://ssrn.com/abstract=1982237

Flanagan, T., & Greenhut, J. (2021). *Adaptive dollar-cost averaging: A new approach*. *Journal of Investment Strategies*, 10(2), 1-15.

Graham, B. (2003). *The intelligent investor*. Harper Business. (Original work published 1949)

Investopedia. (2025). *Lagging indicators*. https://www.investopedia.com/terms/l/laggingindicator.asp

Johnson, M., & Lee, S. (2022). Predicting stock returns using neural networks. *Journal of Financial Data Science*, 4(1), 45-60.

Kirkby, J. L., Mitra, L., & Nguyen, D. (2020). An analysis of dollar-cost averaging in trending markets. *Journal of Behavioral Finance*, 21(4), 352-366. https://doi.org/10.1080/15427560.2020.1746385

Morningstar. (2025). *Dollar-cost averaging in volatile markets*. https://www.morningstar.com/articles/1020304/dollar-cost-averaging-in-volatile-markets

Smith, J., Brown, A., & Wilson, T. (2023). Machine learning in finance: A review. *Financial Analysts Journal*, 79(2), 5-25. https://doi.org/10.1080/0015198X.2023.2180001

Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance*, 62(3), 1139-1168. https://doi.org/10.1111/j.1540-6261.2007.01232.x