Deep Learning for Financial Time Series

(DL)

Kexin Yang(Tahlia)

Username: yangkexinsouffle@163.com

Date: 21/08/2024

Catalog

Instruction 4
Step 1: Problem Statement 6
Step 2 : Feature Engineering 7
1. Price Features 7
2. Financial Features 7
Additional Derived Features 8
Step 3 : Exploratory Data Analysis 9
1. Missing Value Analysis 9
2. Data Integrity and Structure Examination 9
3. Distribution Analysis 9
Step 4 : Cleaning & Imputation 10
Step 5 : Feature Scaling 10
1. Feature Correlation and Reduction 11
2. Standardization of Features 11
3. Clustering for Feature Engineering 11
4. Feature Importance and Selection 12
5. Outcome of the Process 12
Step 6 : Model Building & Optimization (DL) 13
1. Dataset Preparation with Lookback Approach 13
2. Feature and Target Definition 13
3. Data Splitting for Training and Validation 13
4. Neural Network Architecture and Training 13
5. Hyperparameter Optimization 14
Step 7 : Model Evaluation & Backtesting 15
1. Predictive Quality Investigation 15
2. Validation Set Evaluation 15
3. Test Set Evaluation 16
4. Backtesting Applied to Trading Strategy 16
5. Model Evaluation Summary 16
6. Conclusion 17
Reference 17

Instruction

The objective of this research is to predict the short-term directional movement of NVIDIA Corporation's common stock (NVDA). Utilizing daily trading data from August 15, 2019, to August 14, 2024, I aim to discern whether the stock will experience an upward or downward movement on the following trading day. By integrating a broad array of financial and trading metrics into a predictive model, I seek to provide a robust tool for traders and financial analysts to forecast stock movements more accurately.

This investigation not only involves the development of a sophisticated model using historical price data and financial ratios but also focuses on enhancing the model's accuracy through advanced feature engineering. This includes the examination of price movements, trading volumes, and various financial indicators that collectively offer insights into the stock's financial health and market behavior.

My approach is grounded in rigorous data analysis, beginning with a comprehensive exploratory data analysis to identify correlations and potential anomalies within the data. This is followed by meticulous data cleaning, feature selection, and model training, employing modern machine learning techniques to refine my predictive accuracy.

Milestone Summary	Description
Step 1: Problem Statement	The objective of this analysis is to predict the short-term directional movement (up/down trend) of NVIDIA Corporation Common Stock (NVDA), identified by its ISIN US67066G1040. We seek to ascertain whether the stock will experience an upward (denoted by 1) or downward (denoted by 0) movement on the following trading day. Underlying assets: NVIDIA Corporation Common Stock (NVDA) Timeframe: from August 15, 2019, to August 14, 2024 Frequency of data used: daily
Step 2 : Feature Engineering	The features selected for this predictive model include: Price-related metrics: Price, CVol, Change, % Change, % Return, Total Return (Gross), Amount, Open, High, Low. Financial metrics: Gross Margin, SG&A to Sales, Operating Margin, Pretax Margin, Net Margin, Free Cash Flow Margin, Capex To Sales, Return on Assets, Return on Equity, Return on Common Equity, Return on Total Capital, Return on Invested Capital, Cash Flow Return on Invested Capital, Price/Sales, Price/Earnings, Price/Book Value, Price/Tangible Book Value, Price/Cash Flow,

	Price/Free Cash Flow, Dividend Yield (%), Enterprise Value metrics, Debt ratios, Per Share Data, and various asset and liability metrics. Derived metrics: Log_Returns, EWMA_Volatility. Threshold: small positive returns below 0.25% are categorized as negative (0) to refine the predictive accuracy.
Step 3 : Exploratory Data Analysis	Missing Value Analysis Data Integrity and Structure Examination Distribution Analysis
Step 4 : Cleaning & Imputation	Handling of Lookback Window and Delay: For this model, the lookback window is defined as 30 days. Additionally, a delay of 2 days is incorporated into the model. This delay accounts for the lag time between significant market events or financial disclosures and their perceptible effects on stock prices.
Step 5 : Feature Scaling	Feature Correlation and Reduction: Features that are highly correlated (beyond a specified threshold, typically set at 0.9) are identified and considered redundant, as they provide similar information. Standardization of Features Clustering for Feature Engineering: To further reduce dimensionality and to extract new features, K-means clustering and SOM are utilized. Feature Importance and Selection: Finally, to identify the most predictive features, a Random Forest classifier is used to assess feature importance. The top 20 most significant features are selected based on their importance scores, providing a focused subset of features that are most effective in predicting the target variable.
Step 6 : Model Building & Optimization (DL)	Dataset Preparation with Lookback Approach: 1. Feature and Target Definition 2. Data Splitting for Training and Validation 3. Neural Network Architecture and Training 4. Hyperparameter Optimization: • Model Evaluation and

	Performance Metrics: With a particular focus on the Precision metric, which provides a comprehensive measure of model performance across different classification thresholds, which is especially crucial in scenarios where the cost of false positive predictions is considered high. • Visualization of Training Progress: Plots of loss and precision, offering insights into the model's behavior during training, such as identifying overfitting or underfitting patterns, and the effectiveness of the learning rate adjustments. • Final Model Selection
Step 7 : Model Evaluation & Backtesting	Using a comprehensive set of metrics, including the Area Under the Curve (AUC), confusion matrix, classification report, and balanced accuracy.

Step 1: Problem Statement

The objective of this analysis is to predict the short-term directional movement (up/down trend) of NVIDIA Corporation Common Stock (NVDA), identified by its ISIN US67066G1040. We seek to ascertain whether the stock will experience an upward (denoted by 1) or downward (denoted by 0) movement on the following trading day. This study utilizes a dataset procured from the FactSet database, encompassing daily trading data spanning a five-year period from August 15, 2019, to August 14, 2024. By leveraging historical price fluctuations, trading volumes, and a variety of financial ratios, we aim to develop a predictive model that accurately forecasts daily price movements. To enhance the precision of our model, the daily stock data is supplemented with annual financial ratios. For instance, the financial ratio data for January 20, 2020, is aligned with the annual financial data reported for the year 2020, meaning all stock data from the year 2020 will utilize the 2020 annual financial ratios. This approach ensures that the financial ratios applied are contemporaneous with the fiscal periods they represent, thus providing a robust foundation for the predictive analysis. The integration of these features is intended to enhance the model’s predictive accuracy by providing a comprehensive view of the stock’s financial health and market behavior. The dataset's selection, timeframe, and the daily frequency of

the data are deliberately chosen to optimize the relevance and reliability of the predictive outcomes, adhering to the best practices in financial data analysis.

Step 2 : Feature Engineering

The feature selection for this project is bifurcated into two primary categories: Prices and Financials, each providing a unique set of predictors that are instrumental in forecasting the short-term price movements of NVIDIA Corporation Common Stock (NVDA).

1. Price Features

- Price History: Reflects the historical trading prices (close prices), essential for trend analysis.
- Volume (CVol): Indicates the number of shares traded, providing insights into market activity.
- Change: The absolute difference in price between two consecutive days, highlighting daily volatility.
- % Change: This percentage reflects the relative change from the previous day, capturing market sentiment.
- % Return & Total Return (Gross): These metrics measure the returns over a period, adjusted for dividends and splits.
- Amount: Represents the total dollar amount traded, correlating with market liquidity.
- Open, High, Low: These intraday prices offer insights into daily price ranges and potential resistance or support levels.

2. Financial Features

Derived from the company's annual financial statements, these features encapsulate the financial health and operational efficiency of NVIDIA:

- Ratios such as Gross Margin, Operating Margin, and Net Margin: These provide a clear picture of profitability and operational efficiency.
- Liquidity and solvency ratios like Current Ratio and Debt to Equity: Indicators of financial stability.
- Return metrics like ROA (Return on Assets) and ROE (Return on Equity): Gauge the effectiveness of management in generating returns on investments.
- Valuation ratios including P/E (Price to Earnings) and P/B (Price to Book): Assess market expectations and company valuation.
- Per share data such as EPS (Earnings Per Share) and DPS (Dividends Per Share): Offer insights into profitability and shareholder returns on a per-share basis.

Additional Derived Features

- Log Returns: Calculated from consecutive daily prices to capture the natural logarithm of the price ratio, providing a continuous measure of returns.
- EWMA Volatility: Employing an Exponentially Weighted Moving Average with a decay factor (lambda) of 0.94, focusing on the most recent 30 days to capture relevant volatility trends impacting forecast accuracy.
- Handling Initial Missing Data in Derived Features: In the construction of our predictive model, certain features derived from consecutive daily prices, such as Log_Returns and EWMA_Volatility, are crucial for capturing the stock's price dynamics. However, the first trading day in our dataset presents a unique challenge as there is no preceding day to calculate these metrics. To address this, we implement the following data imputation strategy:
    - Log_Returns Imputation: For the first day's Log_Returns, where computation is not possible due to the absence of a prior day's price, we propagate the value from the second day. This approach assumes that the initial change in price is reflective of the second day's dynamics, thus providing a conservative yet informed estimate for the model's input.
    - EWMA_Volatility Imputation: Similarly, the EWMA_Volatility for the first day is filled with the value from the second day. Given that EWMA incorporates a decay factor to weight recent prices more heavily, using the second day's volatility provides an initial estimate that aligns with the short-term trend observed as the data sequence begins.

Label Specification and Class Imbalance Strategy: Given the peculiar nature of NVIDIA's stock movements, small positive returns below 0.25% are categorized as negative (0) to refine the predictive accuracy. This labeling strategy helps address potential class imbalances, ensuring that the model is not biased toward predicting predominantly positive or negative movements.

Comprehensive Feature List: The finalized features for this predictive model include:

- Price-related metrics: Price, CVol, Change, % Change, % Return, Total Return (Gross), Amount, Open, High, Low.
- Financial metrics: Gross Margin, SG&A to Sales, Operating Margin, Pretax Margin, Net Margin, Free Cash Flow Margin, Capex To Sales, Return on Assets, Return on Equity, Return on Common Equity, Return on Total Capital, Return on Invested Capital, Cash Flow Return on Invested Capital, Price/Sales, Price/Earnings, Price/Book Value, Price/Tangible Book Value, Price/Cash Flow, Price/Free Cash Flow, Dividend Yield (%), Enterprise Value metrics, Debt ratios, Per Share Data, and various asset and liability metrics.
- Derived metrics: Log_Returns, EWMA_Volatility.

These extensive features ensure a robust foundation for accurate and insightful prediction of short-term stock price movements, harnessing both historical pricing data and a deep dive into financial health metrics.

Step 3 : Exploratory Data Analysis

As part of a rigorous approach to building a predictive model for NVIDIA Corporation's stock, the Exploratory Data Analysis (EDA) phase is essential for thoroughly understanding the dataset's characteristics and refining our feature selection strategy. EDA is pivotal for identifying correlations, potential redundancies, and anomalies within the data, thereby streamlining the modeling process through effective dimensionality reduction and feature scaling.

1. Missing Value Analysis

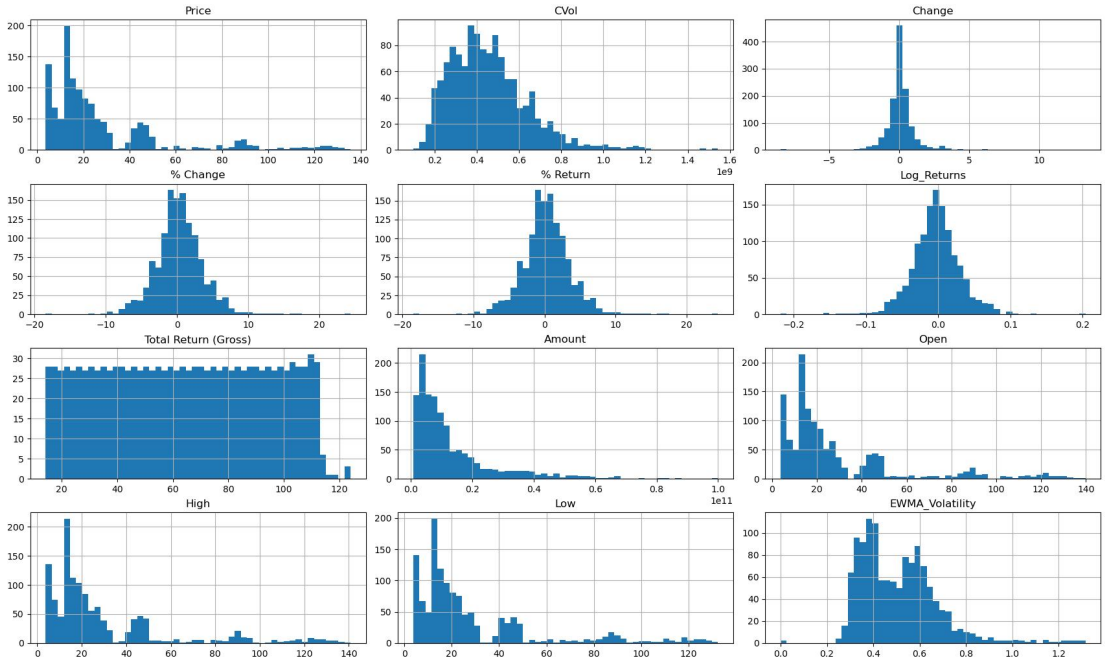
The initial step in our EDA involves an exhaustive examination of the dataset to ascertain the presence of missing values across all columns. This process confirms the completeness of the dataset, ensuring robustness and reliability without the need for additional data imputation.

2. Data Integrity and Structure Examination

We then assess the data types and gather descriptive statistics to ensure data consistency and suitability for further analysis. This analysis aims to verify that each feature is appropriately classified according to its nature (numerical or categorical) and to understand the central tendencies and dispersion within the financial indicators. This step is crucial for tailoring subsequent data transformation techniques and for identifying outliers that may influence the model's predictive accuracy.

3. Distribution Analysis

To visually assess the distribution and behavior of key financial metrics, histograms are utilized for selected features such as Price, CVol, Change, % Change, % Return, Log_Returns, Total Return (Gross), Amount, Open, High, Low, and EWMA Volatility. The objective of this visualization is to discern distribution patterns and detect skewness or anomalies in the features. Understanding the variability and presence of outliers in financial data through histograms is particularly useful, which can significantly impact the model's performance. This graphical analysis assists in determining the appropriate normalization or scaling methods that may be required to align feature scales prior to model training.



Step 4 : Cleaning & Imputation

Ensuring the integrity and accuracy of the data used in predictive modeling is paramount. As such, the data cleaning and imputation phase is critical, especially when using varied datasets, feature sets, and lookback lengths.

Handling of Lookback Window and Delay

For this model, the lookback window is defined as 30 days. This duration has been selected to capture a comprehensive view of short-term trends and cycles in the stock's performance. Historical data spanning this period is utilized to predict the next day's return sign (up/down), providing a robust basis for the model's forecasts.

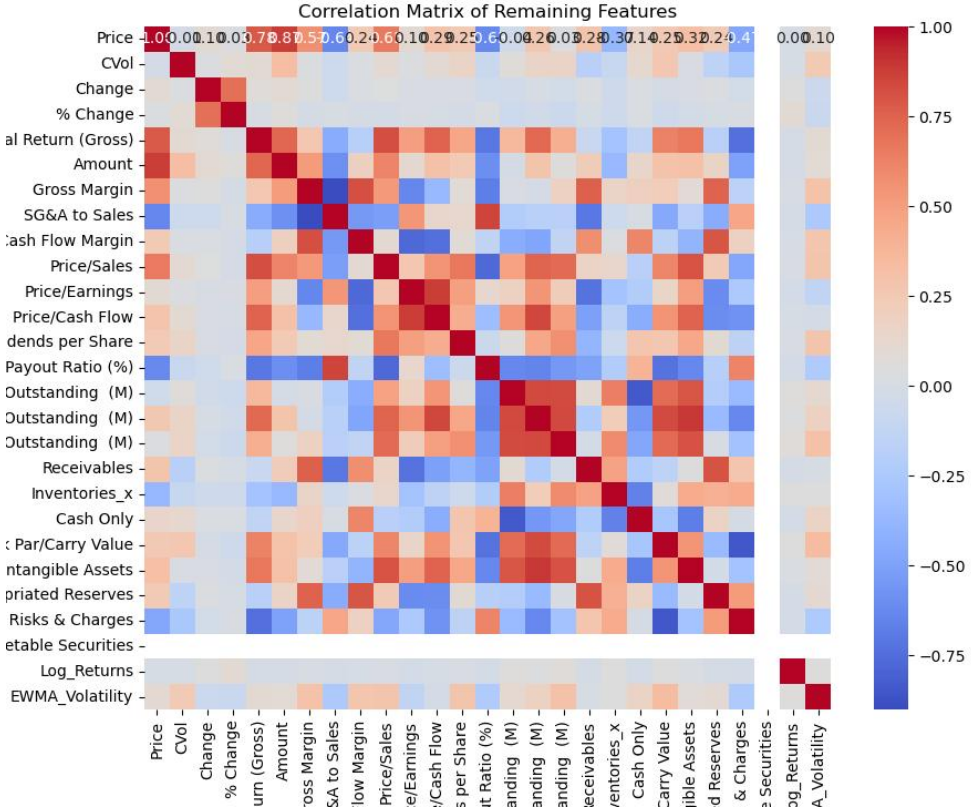
Additionally, a delay of 2 days is incorporated into the model. This delay accounts for the lag time between significant market events or financial disclosures and their perceptible effects on stock prices. The inclusion of this delay helps in capturing the realistic reaction time of the market to new information, ensuring that the model aligns more closely with real-world trading scenarios.

Step 5 : Feature Scaling

In this phase of the modeling process, rigorous feature transformation and dimensionality reduction are applied based on insights gathered during the Exploratory Data Analysis (EDA). These steps are crucial for enhancing model performance and ensuring the generalizability of the predictions.

1. Feature Correlation and Reduction

The initial task involves calculating a correlation matrix for all features to identify the degree of linear relationships between them. This matrix is visually represented through a heatmap, allowing for an intuitive understanding of feature interdependencies. Features that are highly correlated (beyond a specified threshold, typically set at 0.9) are identified and considered redundant, as they provide similar information. These redundant features are subsequently removed from the dataset to avoid multicollinearity, which can adversely affect the model's performance.



2. Standardization of Features

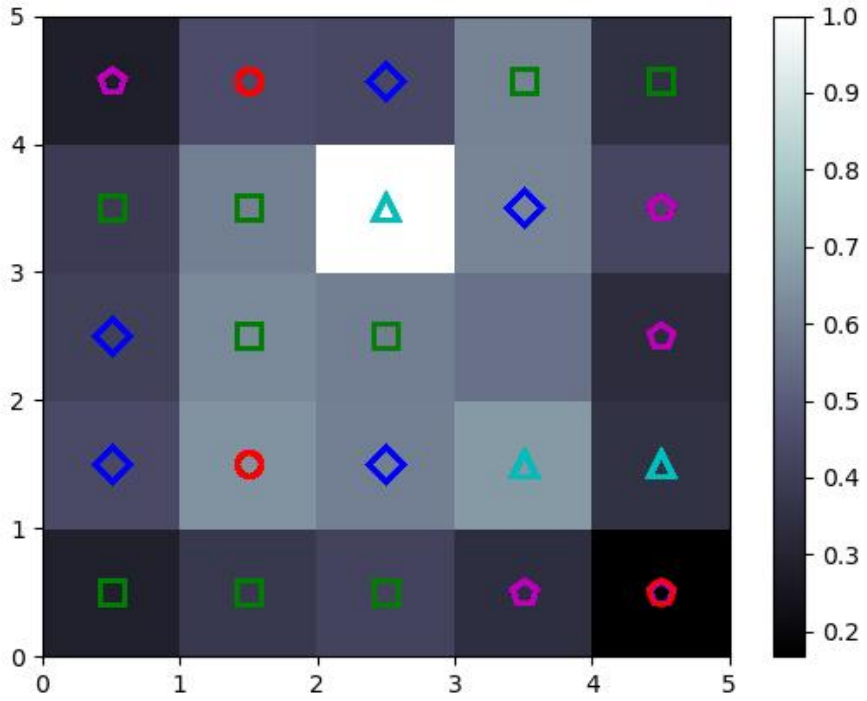
Following the removal of highly correlated features, the remaining features undergo standardization. This process involves scaling the features so that they have a mean of zero and a standard deviation of one. Standardizing the features is essential for many machine learning algorithms that are sensitive to the scale of input data, such as K-means clustering and Self-Organizing Maps (SOM), ensuring that each feature contributes equally to the analysis.

3. Clustering for Feature Engineering

To further reduce dimensionality and to extract new features, K-means clustering and SOM are utilized.

K-means clustering groups the data into clusters based on feature similarity, which can help in identifying inherent groupings within the data. The cluster labels from K-means are then used as new features, potentially capturing non-linear relationships that are not apparent through individual features alone.

SOM Application: Similarly, a Self-Organizing Map is employed to project the high-dimensional data into a lower-dimensional space. The training process involves adjusting the weights of a network to preserve the topological properties of the input space, which aids in visualizing complex patterns in the data. The output grid from SOM provides a condensed representation of the data, highlighting clusters and their relationships, which can be useful for both feature extraction and outlier detection.



4. Feature Importance and Selection

Finally, to identify the most predictive features, a Random Forest classifier is used to assess feature importance. This method offers a straightforward metric to gauge the relative importance of each feature in making accurate predictions. The top 20 most significant features are selected based on their importance scores, providing a focused subset of features that are most effective in predicting the target variable. This feature selection step not only improves the model's performance but also reduces overfitting by eliminating less informative features.

5. Outcome of the Process

The culmination of these steps results in a refined set of features that are scaled, less collinear, and enhanced with new attributes derived from clustering methods. This streamlined feature set is instrumental in building a robust predictive model that is both

accurate and efficient in handling new data. After comprehensive analysis and selection processes, the top 20 most significant features essential for predicting the future price movements of NVIDIA Corporation's stock were determined. The prioritized features, derived from their importance using the Random Forest classifier, are: 'Price', 'CVol', 'Change', '% Change', 'Total Return (Gross)', 'Amount', 'SG&A to Sales', 'Price/Sales', 'Price/Earnings', 'Price/Cash Flow', 'Receivables', 'Inventories_x', 'Cash Only', 'Common Stock Par/Carry Value', 'Intangible Assets', 'Other Appropriated Reserves', 'Log_Returns', 'EWMA_Volatility', 'Return sign', and 'Kmeans_Cluster'.

Step 6 : Model Building & Optimization (DL)

In this critical phase of our analysis, extensive efforts are dedicated to constructing and optimizing a deep learning model for predicting the daily return sign (up/down) of NVIDIA Corporation's stock. This step involves a methodical approach to designing, training, and refining a neural network architecture, ensuring it aligns perfectly with our objectives and data characteristics.

1. Dataset Preparation with Lookback Approach

Initially, the dataset is prepared by integrating a lookback window of 30 days and a delay of 2 days. This structured preparation helps the model capture the necessary temporal dynamics by utilizing past data points to predict future stock movements. The lookback method is instrumental in incorporating historical context, which is crucial for forecasting time-series data.

2. Feature and Target Definition

The features for the model are meticulously selected based on their predictive importance, as identified in previous steps. The target for prediction is the 'Return sign' of the stock, categorizing its movement as up or down. This clear definition of features and target ensures that the model's inputs and outputs are optimally aligned with the forecasting objectives.

3. Data Splitting for Training and Validation

The prepared data is divided into training, validation, and test sets, with proportions of 60%, 20%, and 20% respectively. This segmentation allows for comprehensive training of the model while providing robust datasets for validating and testing the model's performance, ensuring that the model generalizes well to new, unseen data.

4. Neural Network Architecture and Training

The neural network is structured with multiple LSTM (Long Short-Term Memory) layers, known for their efficacy in handling sequential data. The architecture includes:

First LSTM Layer: This input layer consists of 100 LSTM units and is configured to return sequences. This setup ensures that each timestep's output can be passed on to the next layer, capturing temporal dependencies effectively.

Second LSTM Layer: Another layer of 100 LSTM units follows, which does not return sequences but instead provides output only from the last timestep of the sequence. This configuration helps in refining the model's ability to predict based on the learned features from the entire sequence.

The training process is enhanced by an adaptive learning rate, which adjusts based on the validation loss to improve training dynamics and model convergence.

5. Hyperparameter Optimization

Hyperparameter tuning is performed rigorously to find the optimal settings that yield the best predictive performance. Parameters such as learning rate, number of LSTM units, and batch size are adjusted in a systematic way to enhance the model's ability to learn from the data effectively.

- Model Evaluation and Performance Metrics:

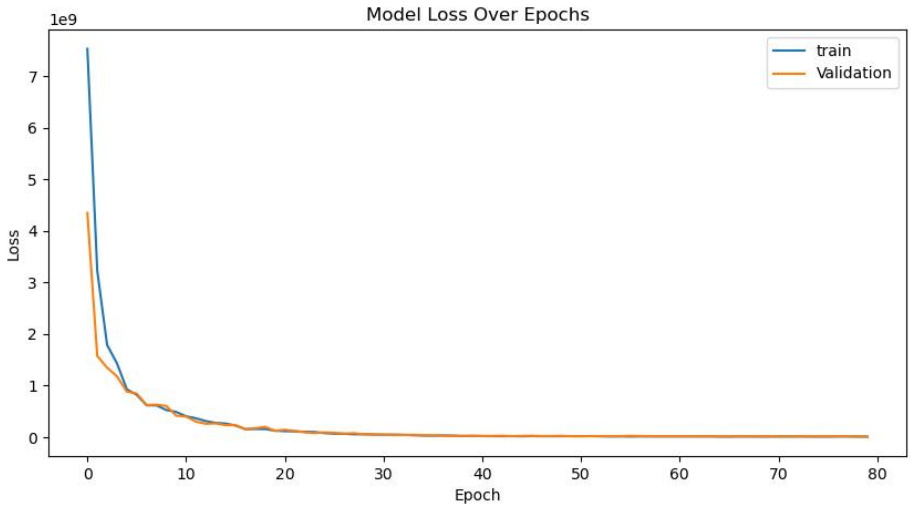
  Post-training, the model is evaluated on both validation and test datasets to assess its predictive accuracy, with a particular focus on the precision metric, which provides a comprehensive measure of model performance across different classification thresholds, which is especially crucial in scenarios where the cost of false positive predictions is considered high.This evaluation helps confirm the robustness of the model before it is finalized.

- Visualization of Training Progress:

  Training and validation metrics are visualized over epochs to monitor the model's learning progress. These visualizations typically include plots of loss and precision, offering insights into the model's behavior during training, such as identifying overfitting or underfitting patterns, and the effectiveness of the learning rate adjustments.

  - Final Model Selection:

  The best-performing model configuration, determined after extensive hyperparameter optimization and comparative analysis with baseline models, is selected. This final model stands as the cornerstone of our predictive analysis, encapsulating the optimal combination of architecture and training dynamics to forecast stock movements effectively.



Step 7 : Model Evaluation & Backtesting

In this final phase of the project, the performance of the deep learning model designed to predict the daily return sign (up/down) of NVIDIA Corporation's stock is rigorously evaluated and backtested. This evaluation not only gauges the model's predictive accuracy but also assesses its practical applicability in a trading strategy.

The performance of the proposed classifier is rigorously assessed using a comprehensive set of metrics, including the Area Under the Curve (AUC), confusion matrix, classification report, and balanced accuracy. These metrics provide a multifaceted view of the model's predictive accuracy and its ability to distinguish between classes under varying thresholds.

1. Predictive Quality Investigation

The model's ability to generate accurate predictions is first examined through its validation performance. Predictions for both the validation and test datasets are generated using the trained model, which then undergo binary thresholding at 0.5 to classify observations as either positive or negative.

2. Validation Set Evaluation

The model's effectiveness on the validation set is quantified by the AUC score, which measures the model's ability to differentiate between the classes without being tied to a specific classification threshold. Additionally, the confusion matrix provides insights into the number of true positives, true negatives, false positives, and false negatives. The classification report offers a detailed account of the precision, recall, and F1-score for each class, while balanced accuracy gives a sense of accuracy adjusted for imbalance in the class distribution. These metrics collectively help in understanding the model's performance nuances and adjusting the model's parameters to optimize performance.

3. Test Set Evaluation

Following the validation process, the model is further evaluated on a separate test dataset to ensure that the model generalizes well to new, unseen data. The same set of metrics used for the validation set is applied to evaluate the test set, ensuring consistency in performance assessment.

4. Backtesting Applied to Trading Strategy

The ultimate test of the model's utility is conducted through backtesting its predicted signals against historical data to simulate real-world trading conditions. This practical application helps to validate the predictive power of the model within the context of market dynamics. The backtesting process focuses on the profitability and risk metrics derived from the model's signals, providing a direct measure of the strategy's potential financial performance.

5. Model Evaluation Summary

Test Set Analysis

On the test dataset, the Precision is slightly higher at 58.39%, reflecting a better ratio of correct positive predictions in new, unseen data.

The AUC Score improved to 0.561, indicating a better but still modest ability to differentiate between the classes at various thresholds.

The Confusion Matrix for the test data [65, 52] for negatives and [56, 73] for positives shows a similar distribution to the validation set, indicating consistency in model performance.

The Classification Report shows an increase in both precision and recall for the positive class on the test set, achieving values of 0.58 for precision and 0.57 for recall.

The Balanced Accuracy of 56.07% on the test set shows an improvement and suggests that the model performs with moderate effectiveness when both classes are considered equally.

```
Test Set Performance:
AUC Score: 0.5614192009540847
Confusion Matrix:
 [[65 52]
 [56 73]]
Classification Report:
              precision    recall  f1-score   support

         0.0       0.54      0.56      0.55       117
         1.0       0.58      0.57      0.57       129

    accuracy                           0.56       246
   macro avg       0.56      0.56      0.56       246
weighted avg       0.56      0.56      0.56       246

Balanced Accuracy: 0.5607235142118863
```

6. Conclusion

These metrics collectively highlight areas of strength and opportunities for improvement in our model. The consistent but moderate performance across both datasets suggests that while the model can identify patterns within the data, there is potential for enhancement, particularly in improving precision and balanced accuracy. Adjustments to model architecture, further feature engineering, or incorporating additional data sources may be considered to boost the model's predictive accuracy and reliability in a real-world trading environment.

Reference

Advanced Machine Learning - I & II, Elective by Kannan Singaravelu, (2021 & 2022)

https://www.cqf.com/about-cqf/program-structure/cqf-qualification/advanced-electives
Short-term stock market price trend prediction using a comprehensive deep learning system by Jingyi Shen & M.

A graph-based CNN-LSTM stock price prediction algorithm with leading indicators by Jimmy Ming-Tai Wu, Zhongcui Li, Norbert Herencsar, Bay Vo & Jerry Chun-Wei Lin, (2021)
https://link.springer.com/article/10.1007/s00530-021-00758-w

CQF Lectures and Python Labs on Machine Learning and Deep Learning, (2024)

Omair Shafiq, Journal of Big Data volume 7, (2020)
https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00333-6