COSE474-2024F: Final Project
“Enhancing Nvidia Stock Price Prediction: A Comparative Study of LSTM and Linear Regression with Sentiment Analysis from FinBERT ”

Muhammad Faiz Chan

1. Introduction

Nvidia stock has been rising to become the second largest company in the United States by market capitalization. In the S&P 500 index, it is also ranked second in the amount of weights it contributed to the index. Thus, a lot of people invest and trade with this stock possibly for long term or short term gains. If someone invested 1,000 US Dollars into the stock 5 years ago with the dividends reinvested, the person would have around 28,000 US Dollars by today which makes predicting its future prices crucial in profit making or minimizing losses.

2. Problem definition & Challenges

The core objective of this project is to harness the power of pre-trained deep learning models, which are typically used in natural language processing (NLP), to predict Nvidia’s stock prices. The aim is to model complex dependencies in both time series financial data and external textual data like financial news and social media. The volatile nature of Nvidia’s stock, coupled with numerous influencing factors such as global events and company-specific news, creates a complex forecasting problem that traditional models struggle to address. Thus, Pre-trained models are used because it’s well-suited for learning patterns, extracting deep contextual features, and can potentially outperform traditional statistical and shallow machine learning methods in understanding financial markets.

Challenges

- External Events: there are a lot of factors that could affect the stock market and one of them are events that are happening in the world such as financial crises or the troubles within the company
- Outliers: because of the same reason as the above, there will be outliers in the dataset as well which resulted in people selling or buying the stock at a fast rate
- Huge Number of Features: Predicting stock prices also involved in understanding the company behind it and what products it sells which could drive the stock prices up or down. Thus, feeding a number of this features in the model would be challenging.

3. Concise Description of Contribution

This project investigates the role of sentiment analysis in predicting stock prices, focusing on integrating FinBert, a pre-trained foundational model tailored for financial sentiment, into predictive modeling frameworks. The primary objective was to evaluate the added value of sentiment analysis when combined with traditional stock market indicators.

To achieve this, FinBert was utilized to extract sentiment scores—positive, neutral, and negative—from financial news and textual data associated with the stock market. These sentiment scores were incorporated into the dataset as additional features, providing a new dimension for analysis. The project then developed two distinct predictive models: a baseline model relying solely on historical stock prices and technical indicators, and an enhanced model that included sentiment data derived from FinBert.

The models were rigorously compared using metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and RPI to quantify their performance. This comparison offered valuable insights into the influence of sentiment analysis on prediction accuracy. While the sentiment-enhanced model demonstrated the potential for greater contextual understanding, it also revealed challenges, such as the risk of overfitting due to the sentiment data’s variability.

To address these challenges, the project explored strategies to mitigate the weight of sentiment features on the predictions. Techniques such as normalizing sentiment values and adjusting feature weights were implemented to balance their contribution, reducing noise while retaining their informative value.

Through this work, the project highlights the interplay between financial sentiment and stock market prediction, offering a nuanced perspective on the benefits and limitations of incorporating pre-trained foundational models like FinBert into financial modeling. This approach underscores the importance of careful feature engineering and evaluation in leveraging advanced machine learning tools for real-world applications.

4. Methods

4.1. Significance and Novelty

This research explores the integration of advanced sentiment analysis with stock price prediction models, addressing the critical question of whether sentiment data enhances predictive accuracy or introduces noise. By employing FinBert, a domain-specific foundational model trained on financial language, the study ensures precise sentiment interpretation, overcoming the limitations of generic models in understanding financial jargon.

The novelty lies in comparing traditional models based solely on historical data with those augmented by sentiment analysis, a dual focus that highlights the impact of combining qualitative and quantitative factors. Additionally, this paper addresses practical challenges, such as overfitting from sentiment data, and proposes methods to balance its influence, improving model robustness.

By bridging advancements in machine learning with finance, this research demonstrates the real-world applicability of foundational models like FinBert, paving the way for interdisciplinary studies and innovation in financial modeling. It provides actionable insights into the role of sentiment analysis, offering significant contributions to the evolution of financial prediction technologies.

4.2. Main Figure

As shown in Figure 1 which uses some part from .(Huang et al., 2023), the main figure shows how the whole model works where the model will use historical data with the moving averages of 20 days, 50 days moving average and the sentiment score which will be generated by using the summaries of news content being used as the input for FinBert where it will output out a sentiment score for that day, as the feature for the LSTM model and the linear regression model.

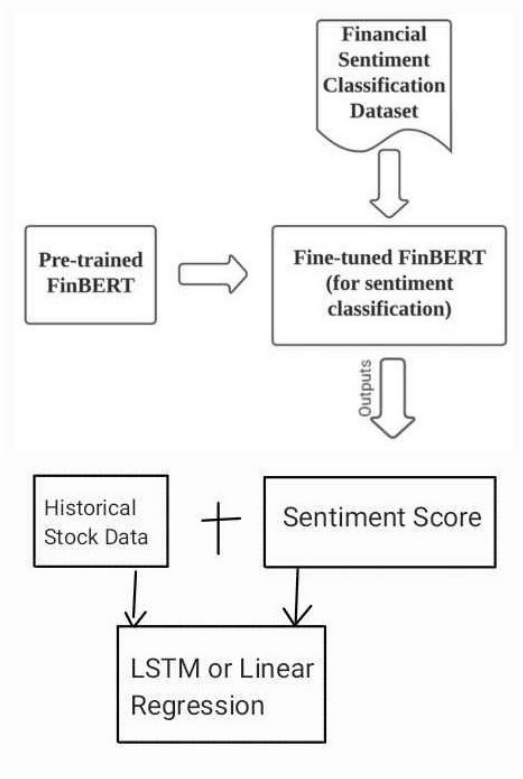
Financial Sentiment Classification Dataset
Pre-trained FinBERT
Fine-tuned FinBERT (for sentiment classification)
Outputs
Historical Stock Data
+
Sentiment Score
LSTM or Linear Regression

Figure 1. Model Architecture.

4.3. Reproducibility

Algorithm 1 Stock Price Prediction with Sentiment Analysis

```
Input: Stock price data $S$, Sentiment data $T$
Output: Predicted stock price $P$
Preprocess Data:
  Load stock price data $S$ and sentiment data $T$
  Merge $S$ and $T$ on corresponding dates
  Filter data for relevant date range and handle missing values
Feature Engineering:
  Calculate technical indicators (e.g., moving averages) from $S$
  Assign sentiment scores to corresponding dates from $T$
Prepare Training and Testing Sets:
  Split data into training and testing subsets (e.g., 80-20 split)
  Normalize features for the LSTM model
Train Linear Regression Model (Baseline):
  Use historical price features and technical indicators
  Fit the model to training data and predict test data
Train LSTM Model:
  Reshape input data to 3D format for LSTM
  Configure LSTM with layers and compile with MSE loss and Adam optimizer
  Train the model using $X_{train}$ and $y_{train}$
Sentiment-Enhanced Prediction:
  Repeat the training process with sentiment scores as additional features
Evaluate Models:
  Compute MSE and MAE for each model
  Compare the performance of models with and without sentiment
```

5. Formulation

5.1. Linear Regression

Linear regression predicts the stock price $\hat{y}$ as:

$$\hat{y} = w_0 + \sum_{i=1}^{n} w_i X_i$$

where X_i are the input features (e.g., technical indicators, sentiment scores), w_0 is the intercept, and w_i are the weights learned by the model. The loss function is defined as:

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

5.2. LSTM

LSTM models process sequential data by computing the hidden state h_t at each time step:

$$h_t = \text{LSTM}(x_t, h_{t-1})$$

where x_t is the input at time t and h_{t-1} is the hidden state from the previous step. The final prediction is given as:

$$\hat{y} = \sigma(W h_T + b)$$

where W and b are learnable weights and biases, and σ is the activation function.

5.3. Sentiment Integration

When incorporating sentiment scores S, the input features become:

$$X_{\text{enhanced}} = \{X, S\}$$

The models are re-trained using these enhanced features.

5.4. Evaluation Metrics

Model performance is evaluated using:

- **Mean Absolute Error (MAE):**

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|$$

- **Mean Squared Error (MSE):**

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

- **Relative Performance Improvement (RPI):**

$$\text{RPI} = \frac{\text{MSE}_{\text{baseline}} - \text{MSE}_{\text{enhanced}}}{\text{MSE}_{\text{baseline}}} \times 100\%$$

6. Datasets

Datasets about daily Nvidia stock prices from kaggle that are a few months before so that we can compare the accuracy of the model against the real thing.

7. State-of-the-art methods and baselines

In this section, we will establish baseline models such as ARIMA, LSTM, and Random Forest, and compare them to the latest state-of-the-art methods, including TFT, Informer, and FinBERT.

8. Datasets

The dataset used includes NVIDIA stock price data from Kaggle, comprising features like Open, Close, High, Low, Volume, and sentiment scores derived using FinBERT. Sentiment data was preprocessed by aggregating daily scores. The stock price features were normalized, and missing values were addressed by interpolation. The data was split into training and testing sets (70/30), ensuring temporal consistency.

It also includes data of news regarding Nvidia from Kaggle, comprising of features like Date, Summary and sentiment analysis but the sentiment analysis was removed to give way for FinBert analysis instead. The summary for the news is in Portugese and to solve this the summary data was translated to English using google translate and the translated data is fed to FinBert to be analyzed for sentiment analysis.

9. Computing Resources

9.1. Cloud-based Resources

- Platform - Google Colab
- CPU/GPU - Tesla T4 GPU with 12.7GB ram
- Virtual Machine Configuration - Configured by the platform; specific details not directly accessible
- Programming Language - Python 3
- Libraries - Pandas, NumPy, scikit-learn, TensorFlow, Seaborn, Kaggle, Google Colab

10. Experimental Design/Setup

10.1. Data Collection

The study utilized two main datasets: historical stock data from Kaggle and sentiment analysis data which had its original sentiment removed and the summary translated to English which were fed to FinBERT, a pre-trained foundational model to output the sentiment score.

The Kaggle dataset included stock market indicators such as Open, Close, High, Low, Volume, and Moving Averages MA 20 and MA 50. The FinBERT sentiment data provided daily sentiment scores, categorized into Positive for 1, Neutral for 0, and Negative for -1, based on financial news and market-related textual data. These datasets were merged on a date basis to create a comprehensive dataset for analysis.

10.2. Data Processing

Data preprocessing was essential for preparing the stock market and sentiment analysis datasets. Missing values in the stock market data were handled using forward filling, followed by the removal of any remaining NaN values. The sentiment data, obtained from the FinBERT model, was then aligned with the stock market data based on shared dates, allowing for direct comparisons between market movements and sentiment trends.

Feature engineering added moving averages (MA 20, MA 50) and sentiment scores as new features to capture market trends and sentiment influence. Additionally, numerical features were normalized to ensure consistent scaling across the dataset, which is crucial for model performance. These preprocessing steps ensured that both datasets were clean, aligned, and ready for predictive modeling.

10.3. Feature Engineering

To improve the predictive capabilities of the models, several new features were engineered from the original stock market and sentiment data. Moving averages (MA 20, MA 50) were calculated to capture trends in stock prices over different time intervals. Sentiment scores derived from the FinBERT model were incorporated as additional features, providing insights into market sentiment.

Additionally, to enhance the model's understanding of market conditions, a normalized sentiment score was created. This normalization ensured that sentiment data had a balanced impact on model predictions, preventing overfitting due to excessively high sentiment values. These engineered features were designed to capture both technical market trends and sentiment-driven factors, thereby improving the model's predictive performance.

10.4. Training and Evaluation

The LSTM model's configuration are as follows:

- Architecture: 1 LSTM layer with 50 units, followed by a dense layer with a single neuron for regression
- Optimizer: Adam with a learning rate of 0.001
- Loss Function: Mean Squared Error (MSE)
- Batch Size: 32
- Epochs: 100
- Early stopping with patience of 10 epochs based on validation loss

The model is run first without the sentiment data using linear regression and LSTM models. The results for the model without the sentiment shows extremely high MAE and MSE value which are 414.10 and 15.32 respectively

However, after implementing the sentiment score, the results improved a lot as can be seen in Figure 2 where the MAE and MSE value reduced to 54.92 and 5.57 respectively. This shows an RPI of MSE to be 86.74 percent and for MAE to be 63.69 percent.

10.5. Result Visualization

The result for the without the sentiment and with sentiment data can be seen as shown in Figure 3 and 4 respectively.

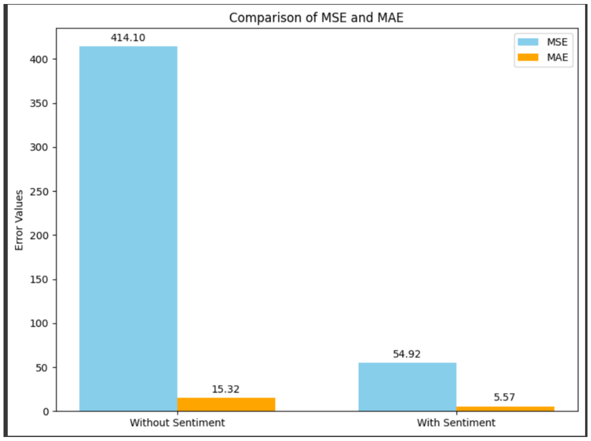

Figure 2. MAE and MSE Result.

The data for the stock price starts from the date 2017-06-29 until 2024-12-03

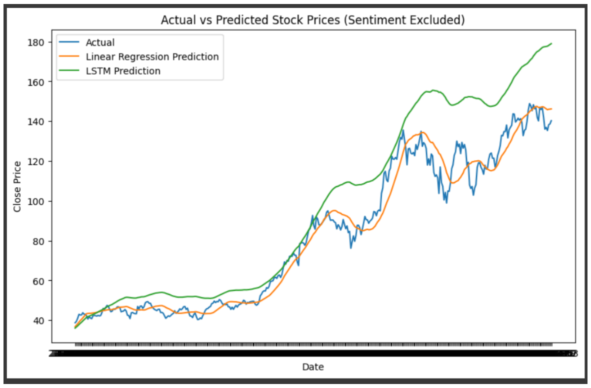

Figure 3. Result Without Sentiment.

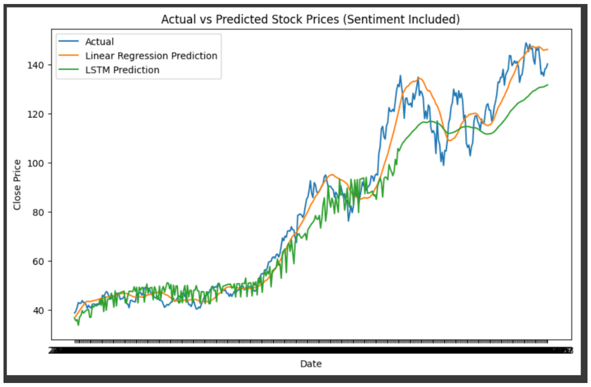

Figure 4. Result With Sentiment.

References

Huang, A. H., Wang, H., and Yang, Y. Finbert: A large language model for extracting information from financial text. *Contemporary Accounting Research*, 40(2):806–841, 2023.