**Stock Price Forecasting using Machine Learning**

Jaya Chandra Kadiveti

Kruthik Reddy Theepireddy

Girish Reddy Maligireddy

School of Professional Studies, Clark university

MSDA 3999: CAPSTONE PRACTICUM

Dr. Khald Aboalayon

**1. Introduction**

**1.1 Overview**

Leveraging machine learning techniques for forecasting future stock prices has emerged as a crucial tool for investors and traders alike. The volatile nature of financial markets presents challenges in accurately predicting stock prices, making the utilization of advanced predictive models imperative. This report outlines the development and implementation of a stock price forecasting model, aimed at providing investors and traders with a reliable tool to navigate market uncertainties.

**1.2 Objective**

The primary objective of this project is to develop a robust machine learning model capable of forecasting future stock prices effectively. By harnessing real-time stock data and employing various predictive algorithms, the aim is to overcome the inherent challenges associated with stock price forecasting. Through rigorous analysis and experimentation, the goal is to identify the most accurate model for predicting stock price movements.

**1.3 Process**

The process of developing the stock price forecasting model entails several key steps:

1. **Data Collection**: Real-time stock data is collected from reliable sources to ensure the accuracy and relevance of the dataset.
2. **Exploratory Data Analysis (EDA)**: A comprehensive EDA is conducted to gain insights into the underlying patterns and trends in the stock data. This step involves visualizing the data, identifying outliers, and understanding the relationships between different variables.
3. **Model Implementation**: Various machine learning models are implemented and evaluated to determine their efficacy in forecasting stock prices. These models encompass a range of algorithms, including but not limited to regression, time series analysis, and ensemble methods.
4. **Model Selection**: The performance of each model is assessed based on predefined metrics such as accuracy, precision, and mean squared error. The model that demonstrates the highest level of predictive accuracy is selected as the optimal choice for forecasting stock prices.
5. **Visualization**: The final step involves plotting the forecasted stock prices using the selected model. A line chart is generated to visually represent the predicted price movements over a specified time horizon.

By following this systematic approach, we aim to develop a reliable and accurate stock price forecasting model that can assist investors and traders in making informed decisions in the financial markets.

**2. Problem Statement**

The accurate estimation of future stock prices has long been a challenging endeavor within the realm of financial forecasting. Traditional methods employed for this purpose often fall short in capturing the intricate dynamics of the market, resulting in unreliable forecasts. This discrepancy poses significant challenges for finance professionals who rely on accurate predictions for strategic decision-making.

* **Complex Market Dynamics**: The stock market is characterized by a myriad of factors and variables that influence price movements, making it inherently complex to model accurately.
* **Limitations of Traditional Methods**: Conventional forecasting methods, such as time series analysis and fundamental analysis, often fail to account for the dynamic and non-linear nature of market behavior, leading to inaccurate predictions.
* **Unreliable Forecasts**: The inability of traditional methods to adequately capture market dynamics results in forecasts that are often unreliable and may lead to suboptimal investment decisions.

**3. Literature Review**

**3.1 Stock Closing Price Prediction using Machine Learning Techniques**

A study conducted by Mehar Vijh et al. in 2020 explored the application of machine learning techniques for predicting stock closing prices. The research focused on utilizing the Random Forest algorithm, a popular ensemble learning method known for its robustness and flexibility.

A screenshot of a computer

Description automatically generated

Fig. 1. Predicted v/s original (expected) closing stock price using RF.

**3.1.1 Methodology**

* **Algorithm Selection:** The study opted for the Random Forest algorithm due to its ability to handle large datasets with high dimensionality and nonlinear relationships effectively.
* **Data Utilization:** Close values of stock prices were used both for training the model and for making predictions. By focusing on this specific aspect of stock data, the model aimed to capture the most relevant information for forecasting future closing prices.
* **Performance Evaluation:** The performance of the predictive model was assessed using the Root Mean Squared Error (RMSE) metric. The reported RMSE values ranged from 1.10 to 3.30, indicating the model's ability to accurately predict stock closing prices within a certain margin of error.
* **Data Splitting Technique:** The study employed a windowing method for splitting the data into training and testing sets. This technique involves dividing the dataset into sequential windows, with each window containing a subset of the data for training and validation purposes.

A screenshot of a computer

Description automatically generated

Fig. 2. RMSE values obtained by Random Forest

**3.2 Stock Market Predictions with LSTM in Python**

Avijeet Bishwal’s work in 2024 delves into the utilization of Long Short-Term Memory (LSTM) networks for stock market predictions, showcasing advancements in machine learning techniques for financial forecasting.

A graph of a graph

Description automatically generated with medium confidence

Fig. 3. Predicted v/s original (expected) closing stock price using LSTM.

**3.2.1 Methodology**

* **Model Selection:** LSTM, a type of recurrent neural network (RNN), was employed for its ability to effectively capture temporal dependencies in sequential data, making it well-suited for time series forecasting tasks.
* **Feature Representation:** The model predicted future stock prices by leveraging past price information. Specifically, it utilized the past 'n' time frames to predict the next 'n' time frames, thereby capturing the temporal dynamics of stock price movements.
* **Input Data:** Unlike the previous study, which used the closing prices, this approach utilized the average of the low and high prices for both training and prediction. This choice of input data likely aimed to incorporate a broader range of information from the stock price history.
* **Performance Evaluation:** The performance of the LSTM model was assessed using the Root Mean Squared Error (RMSE) metric. Reported RMSE values fell within the range of 0.6 to 1.2, indicating relatively low prediction errors and demonstrating the effectiveness of the model in forecasting stock prices.

**4. Methodology**

**4.1 Data Collection**

For the data collection phase of the project, Yahoo! Finance served as the primary source of financial data, offering a comprehensive range of information including stock quotes, financial reports, and market commentary.

**4.1.1 Data Source**

Yahoo! Finance: This platform provided access to real-time stock market data, allowing for the retrieval of historical stock price information and other relevant financial metrics.

**4.1.2 Procedure**

* **Installation and Package Import:** The project involved installing and importing packages from Yahoo! Finance to facilitate the retrieval of stock market data within the Python environment.
* **Stock Selection:** A single stock was chosen for analysis, with its corresponding coordinate values extracted from the desired date range up to the present day. This approach ensured a focused analysis on a specific stock, enabling detailed examination of its historical performance and forecasting potential.
* **Online Data Import:** Unlike traditional datasets, which are pre-existing and static, the project dynamically imported stock market data from Yahoo! Finance in real-time. This dynamic approach allowed for the incorporation of the latest market information into the analysis, enhancing the accuracy and relevance of the forecasting model.

**4.2 Exploratory Data Analysis (EDA)**

Fig. 4. Need of EDA

**4.2.1 Understanding Outliers in Volume:**

* **Volume:**The number of shares of a stock traded within a given period.
* **Outliers:**Extreme data points that fall significantly above or below the typical volume range for a stock. In the image, outliers are the individual dots beyond the top and bottom.
* **AAPL, MSFT, NFLX, EBAY, TSLA:**These stocks show outliers on the high end, indicating unusually heavy trading days at some point in the data period

**A group of graphs with green and black lines

Description automatically generated**

Fig. 5. Box plots of five companies chosen

**4.2.2 Moving Averages:**

**A graph with numbers and lines

Description automatically generatedA graph with blue lines and numbers

Description automatically generated**

**A graph of a price

Description automatically generated with medium confidenceA graph with blue lines and numbers

Description automatically generated**

**A graph with blue and orange lines

Description automatically generated**

Fig. 6. Plots for Moving Averages

**4.2.3 Correlation Analysis:**

* **Purpose:** A correlation matrix shows the strength of the relationship between multiple variables. In this case, the variables are various stock market metrics.
* **Values:** The numbers in the matrix range from -1 to 1.
  1. **-1:** Perfect negative correlation (variables move in opposite directions)
  2. **0:** No correlation (variables aren't related)
  3. **1:** Perfect positive correlation (variables move in the same direction)

A red and blue chart

Description automatically generated

Fig. 7. Correlation Matix for all Parameters

**4.3 Statistical/ Machine Learning**

This Process includes conducting Principal Component Analysis (PCA), Splitting training ad testing data, assigning independent and dependent data for their respective variables before training the model, Training the model, extracting the predictions from the model, and lastly plotting the predictions in a line chart with usual data included.

**4.3.1 Principal Component Analysis (PCA)**

Principle Component Analysis (PCA) was employed due to the high correlation observed among parameters. This technique facilitated the mitigation of multicollinearity issues by transforming the original variables into a set of uncorrelated principal components, thus simplifying interpretation while preserving most of the relevant information.

A graph showing the growth of a company

Description automatically generated A graph showing the growth of a company

Description automatically generated

Fig. 8. Simultaneous difference between before and after implementing PCA

PCA's computational efficiency made it suitable for handling high-dimensional datasets, ensuring robustness in our analysis results. This approach effectively addressed parameter correlation concerns, enhancing the overall quality and interpretability of our findings.

**4.3.4 Data Splitting**

The data is split using Windowing method which is a very effective method commonly used for time series data.

**4.3.4.1 Windowing Method in Machine Learning**

The windowing method is a technique commonly employed in time series analysis and machine learning tasks dealing with sequential data. It involves dividing the sequential data into overlapping or non-overlapping segments called windows. These windows are then used as input samples for training predictive models or for extracting features relevant to the task at hand.

The process of windowing can offer several advantages in machine learning applications:

1. **Temporal Context Preservation**: By dividing the sequential data into windows, the temporal context within each segment is preserved. This allows machine learning models to capture patterns and dependencies over time, which can be crucial for accurate predictions in tasks such as forecasting or anomaly detection.

A computer screen shot of a diagram

Description automatically generated

Fig. 9. Data Splitting method

1. **Feature Extraction**: Windowing facilitates the extraction of relevant features from the sequential data. Each window serves as a sample from which various statistical or domain-specific features can be computed. These features can then be used as input to machine learning algorithms for classification, regression, or other tasks.
2. **Data Augmentation**: In some cases, windowing can be used as a form of data augmentation to increase the size of the training dataset. By generating multiple overlapping windows from the original data, the model is exposed to diverse perspectives of the underlying patterns, potentially improving its generalization performance.
3. **Efficient Processing**: Dividing the data into windows can enhance computational efficiency, particularly when dealing with large datasets. Instead of processing the entire sequence at once, models can be trained on smaller, manageable segments, reducing memory requirements and speeding up training times.

**4.3.3 Models**

The machine learning models used in the research are four namely:

* **Linear Regression**

1. **Simplicity:** Highlight its ease of understanding and implementation compared to more complex models.
2. **Interpretability:** Emphasize how coefficients reveal how factors like earnings influence stock prices.
3. **Identifying Trends:** Linear regression excels at capturing linear or close-to-linear trends in historical data.
4. **Baseline Model:** It serves as a benchmark to compare the effectiveness of more sophisticated models.

A graph showing a line

Description automatically generated with medium confidence

Fig. 10. Predictions by Linear Regression

* Autoregressive Integrated Moving Average (Arima)

1. **ARIMA Components**: ARIMA stands for Autoregressive Integrated Moving Average. It combines three components:
   1. Autoregressive (AR): Incorporates past values of the variable being predicted.
   2. Integrated (I): Uses differencing to make the time series stationary.
   3. Moving Average (MA): Considers the error terms from past predictions.
2. **Time Series Analysis**: ARIMA is effective for analyzing and forecasting time series data, such as stock prices, which exhibit patterns over time.
3. **Forecasting**: Once the ARIMA model is trained on historical data, it can be used to forecast future stock prices based on patterns identified in the data.

A graph showing a line graph

Description automatically generated with medium confidence

Fig. 11. Predictions by ARIMA

* Long Short-Term Memory (LSTM)

1. LSTM is a type of recurrent neural network (RNN) architecture designed to overcome the limitations of traditional RNNs in capturing long-term dependencies in sequential data.
2. LSTM-based models have demonstrated remarkable performance in forecasting stock prices by leveraging historical price data, trading volumes, and other relevant features.
3. They can capture complex patterns and dependencies in stock price movements, including short-term fluctuations and long-term trends.
4. **Advantages:**
   1. **Long-Term Dependencies:** LSTM can capture dependencies over long sequences, making it well-suited for modeling the dynamic nature of stock markets.
   2. **Robustness:** LSTM networks are robust to vanishing gradient problems, allowing for more stable training on large datasets.
   3. **Flexibility:** LSTM architectures can be customized with additional layers, regularization techniques, and attention mechanisms to enhance performance.
   4. LSTM stands as a powerful tool for stock price prediction, offering the capability to capture intricate patterns and adapt to the dynamic nature of financial markets.
   5. Understanding its mechanisms and optimizing its architecture can lead to improved forecasting accuracy and informed investment decisions.

A graph showing the growth of a company

Description automatically generated

Fig. 12. Predictions by LSTM

* Gated Recurrent unit (GRU)

1. GRU is a type of recurrent neural network (RNN) architecture designed to address the vanishing gradient problem and improve the learning of long-term dependencies in sequential data.
2. Unlike LSTM, which has separate memory cells and control gates, GRU combines these into a single mechanism, resulting in a more streamlined architecture.
3. GRU has fewer parameters and computations compared to LSTM, making it computationally more efficient.
4. **Applications:**
   1. GRU is widely used in natural language processing tasks such as machine translation, sentiment analysis, and text generation.
   2. It's also effective for time-series prediction tasks like stock price forecasting, where capturing temporal dependencies is crucial.
5. **Advantages:**
   1. **Simplicity:** GRU has a simpler architecture compared to LSTM, making it easier to understand and implement.
   2. **Efficiency:** Due to its fewer parameters, GRU is computationally more efficient, making it suitable for large-scale applications.
   3. **Performance:** While not always outperforming LSTM, GRU often achieves comparable results with fewer computational resources.

**5. Results**

**5.1 Plots:**

Chosen 5 well known company's stocks to test the model, it goes like:

A graph showing a line graph

Description automatically generated with medium confidence**A graph showing a line graph

Description automatically generated with medium confidenceA graph showing a line

Description automatically generated with medium confidence**A graph showing a line

Description automatically generated**A graph showing a line graph

Description automatically generated with medium confidence**

**Fig. 13.** Predictions by GRU

**5.2 RMSE**

To evaluate the effectiveness of the models, a comparison is made between the two techniques on five different sector companies namely, Microsoft, Apple, Tesla, Meta, Amazon using Linear regression, ARIMA, LSTM, GRU models. Predicted closing prices are subjected to Root Mean Square Error (RMSE) = sqrt [(Σ(Pi – Oi)²) / n].

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Company** | **Data** | **Linear regression** | **ARIMA** | **LSTM** | **GRU** |
| **Apple** | **Training set** | **22.2606** | **15.82** | **0.2183** | **0.2004** |
| **Testing set** | **16.4470** | **16.2** | **0.1517** | **0.1487** |
| **Tesla** | **Training set** | **65.5160** | **43.4171** | **0.1123** | **0.1101** |
| **Testing set** | **58.4098** | **52.7421** | **0.1932** | **0.1713** |
| **Microsoft** | **Training set** | **33.3502** | **19.9436** | **0.08918** | **0.0910** |
| **Testing set** | **42.1783** | **23.2365** | **0.1432** | **0.1375** |
| **Meta** | **Training set** | **34.4927** | **68.7713** | **0.1820** | **0.1677** |
| **Testing set** | **106.8925** | **132.8243** | **0.2496** | **0.2284** |
| **Amazon** | **Training set** | **21.2181** | **18.3560** | **0.1166** | **0.1094** |
| **Testing set** | **32.928** | **24.2342** | **0.1703** | **0.1607** |

**Table 1. RMSE comparison between all five companies**

**6. Lessons Learned & Future Work**

**6.1 Lessons Learned:**

* **RNN, s are the better models to Forecast the stock patterns.**
* **Stock Forecasting Accuracy increases proportionally to the number of factors considered.**

**6.2 Future Work:**

* **Data Enhancement: Using macroeconomic indicators like GDP, Inflation etc**
* **Risk Management: Integrate risk management techniques into the forecasting framework, such as Value at Risk (VaR) analysis or stress testing.**

**7. Conclusion**

Though no model cannot guarantee you with the exact predictions. It can help investors and traders amplify investment strategies and better risk management practices in the financial markets.

**References:**

* Stock Closing Price Prediction using Machine Learning Techniques (Mehar Vijh et al.,2020)
* Stock Market Predictions with LSTM in Python. (Avijeet Bishwal, 2024)