Integrative Machine Learning Approaches for Stock Price Prediction with Apple Inc. and S&P 500 Datasets

Poojith Sonti (02080247)

Abstract

The project stock price prediction presents a comprehensive analysis aimed at predicting stock prices, leveraging historical data sets to model and forecast future market behaviors. By dissecting two distinct data sets—one featuring a wide array of stocks over an extended period, and another focusing on Apple Inc. over a specific year—this research employs statistical and machine learning techniques to derive insights into stock price movements.

The first part of the analysis involves a dataset containing adjusted close prices, which provides a macroscopic view of the market and allows for the identification of long-term trends and anomalies across various stocks. This segment of the study focuses on understanding the effects of market-wide events and the dynamics of price adjustments.

In the second part, a detailed examination of AAPL's daily trading figures such as open, close, high, low, and volume offers a microscopic look at the factors influencing an individual stock. The intent is to explore patterns, seasonal effects, and other influences specific to AAPL, providing a granular perspective on its stock behavior.

Advanced predictive models, including time series analysis and machine learning algorithms, are applied to forecast future price movements with a goal of achieving high accuracy in predictions. These models are evaluated based on their predictive performance and practical applicability in real-world trading scenarios.

The culmination of this project is the development of a predictive tool that aids investors in making informed decisions, optimizing their investment strategies, and mitigating risks associated with stock trading. The insights gained from both datasets are synthesized to offer a robust understanding of both specific stock trajectories and broader market trends, highlighting the complex interplay of factors that drive stock market dynamics.

Motivation

I chose this project on stock price prediction to harness the potential of data science and machine learning in financial markets, an area ripe for technological innovation. The fluctuating nature of stock prices presents a significant challenge and an opportunity to apply analytical techniques for better investment decisions. My motivation is driven by the desire to understand the underlying patterns and factors that influence market movements, enabling more accurate forecasts. This project also allows me to explore the practical application of theoretical concepts learned in data analytics, particularly in a real-world scenario like the stock market. Moreover, the ability to

potentially contribute tools that assist investors in navigating the complexities of financial investments excites me, as it combines my interests in finance and technology to create impactful solutions.

1. Introduction

In the ever-evolving landscape of finance, the ability to predict stock prices with high accuracy remains a pivotal challenge and a significant opportunity for investors worldwide. This project aims to develop a robust stock price prediction model using machine learning techniques, focusing primarily on the historical stock data of Apple Inc. (AAPL). The predictive model leverages various algorithmic strategies and a comprehensive dataset that includes daily trading information such as opening and closing prices, high and low prices, and volume of stocks traded. By analyzing trends and patterns from past stock movements, the model seeks to forecast future prices, offering valuable insights that could potentially lead to informed investment decisions.

1.1 Significance: The significance of this project lies in its ability to transform raw financial data into actionable insights, which is crucial in the high-stakes environment of stock trading. Accurate stock price predictions can lead to substantial economic gains and are vital for portfolio management, risk management, and strategic planning. In the broader context, enhancing the predictive accuracy of stock prices supports the financial industry's stability by providing investors and policymakers with tools to better anticipate market movements. This project not only aims to contribute to academic knowledge in financial econometrics and machine learning but also strives to bridge the gap between theoretical financial models and their practical applications.

1.2 Contribution: This project contributes to the field of financial analytics by implementing and comparing several machine learning techniques to determine the most effective method for predicting stock prices. The use of a detailed dataset comprising Apple Inc.'s stock prices allows for a focused approach to understanding the dynamics of tech industry stocks. Additionally, the project introduces a novel integration of data preprocessing techniques and feature engineering to enhance model performance. By sharing findings and methodologies through comprehensive documentation and open-source code, this work encourages replication and further exploration in both academic and practical domains.

1.3 Impact: The potential impact of this stock price prediction project is multifaceted. For the financial sector, it offers a tool that can enhance investment strategies and improve the accuracy of portfolio forecasts, directly influencing profitability. On an academic level, it provides a case study for the application of advanced machine learning techniques in a real-world scenario, contributing to educational pursuits in data science and finance. Socially, improved predictive models can lead to more stable markets by mitigating risk and reducing the frequency of economic shocks caused by volatile stock movements. Ultimately, the project aims to empower individual investors and large institutions with improved decision-making capabilities, promoting a more informed and efficient marketplace.

2. Literature review

2.1 Current State: Recent advancements in machine learning have significantly influenced the field of financial forecasting, particularly in stock price prediction. Core developments include the application of various neural network architectures, such as Long Short-Term Memory (LSTM) networks, which are adept at capturing temporal dependencies in time-series data like stock prices. Researchers have also extensively utilized Convolutional Neural Networks (CNNs) to process price movements as image-like structures, providing a unique perspective on pattern recognition in financial markets. Furthermore, the integration of Graph Neural Networks (GNNs) has been explored to understand the interconnected nature of financial entities and market indices. Innovations such as attention mechanisms have been introduced in models like the Transformer, enhancing the ability to focus on crucial parts of data sequences for better prediction accuracy. These methods have been pivotal in advancing tasks that require nuanced understanding of market dynamics, such as algorithmic trading and risk assessment.

2.2 Gap Identification: Despite these advancements, several gaps remain in the literature on stock price prediction, which this research seeks to address. A primary concern is the volatility and non-linear nature of stock markets, which many models fail to adequately capture, often leading to inaccuracies in volatile market conditions. Additionally, the overfitting of models to past market behaviors without considering the influence of unforeseen macroeconomic changes poses a significant challenge, resulting in predictions that may not hold under future market conditions. There is also a lack of integration between different types of data, such as textual news and fundamental analysis, which can provide crucial contextual insights that affect stock prices. Moreover, the computational efficiency of current models often remains unoptimized for real-time trading, a vital capability for practical deployment in financial markets. Our research addresses these gaps by incorporating advanced anomaly detection techniques to better model market volatility, enhancing models with hybrid approaches that integrate multiple data sources, and optimizing computational strategies to enable real-time predictive capabilities. Through these innovations, our work not only tackles the identified shortcomings but also pushes the frontier of what is possible in stock market prediction models, setting a new standard for accuracy and efficiency.

3. Methodology

The methodology adopted for the analysis and prediction of stock prices involves a series of systematic steps designed to ensure the accuracy and reliability of the predictive models. The process encompasses data preparation, exploratory data analysis, feature engineering, model training, and evaluation. Herein, we detail each step involved.

1. Importing Libraries and Loading Data

The initial phase begins with setting up the computational environment, where essential libraries such as Pandas, NumPy, Seaborn, and Matplotlib are imported. These libraries facilitate data manipulation, statistical analysis, and graphical representation. The stock market data, typically

comprising fields like Date, Open, High, Low, Close, and Volume, is loaded from a CSV file into a Pandas DataFrame. This structure supports efficient data handling and manipulation.

2. Data Exploration and Cleaning

Upon loading the data, preliminary data exploration is conducted using methods like df.head() to view the first few entries and df.describe() to obtain descriptive statistics that highlight the central tendencies and dispersion in the data. This phase is critical for identifying missing values or anomalies. Cleaning the data involves handling missing values either by imputation—replacing them with the mean of the respective column—or by omitting the incomplete rows entirely. Moreover, irrelevant features might be removed to streamline the analysis.

3. Correlation Analysis

To understand the relationships between variables, a correlation matrix is computed, which provides Pearson correlation coefficients. This analysis helps in pinpointing the features that have significant relationships with the target variable, which in this case is the 'Close' price of the stock. A heatmap visualization of the correlation matrix further aids in identifying highly correlated variables visually, enabling informed decisions on feature selection.

4. Feature Engineering and Data Normalization

Feature engineering involves selecting key features that are strongly correlated with the target variable to reduce dimensionality and enhance model performance. Post feature selection, data normalization is implemented using the MinMaxScaler. This step ensures that all feature values have a uniform scale, a prerequisite for optimal performance in many machine learning algorithms due to their sensitivity to the variance in data scales.

5. Model Preparation and Training

Multiple regression and neural network models, including Linear Regression, ANN, CNN, and LSTM, are prepared for training. These models are chosen to cater to different aspects of time-series prediction in stock market analysis. Training involves configuring various parameters such as the number of epochs, batch size, and specifics of neural network architecture (like the number of layers and activation functions). The models are then trained on the prepared dataset.

6. Evaluation and Visualization of Results

Post-training, models are evaluated using metrics such as the R^2 score and Mean Squared Error (MSE) to quantify their accuracy. The final step involves visualizing the predictions from each model against the actual stock prices using line plots. This visual comparison is crucial for assessing the effectiveness of each model in capturing the trends and fluctuations in stock prices.

In conclusion, the methodology adopted encompasses a comprehensive approach to data analysis and modeling, ensuring a robust framework for predicting stock prices. The iterative process of training and evaluation helps in refining the models to achieve high accuracy, making them valuable tools for investors and analysts alike in making informed decisions.

Now we elaborate on the methodologies employed to predict Apple's stock prices using two distinct approaches: a Random Forest Regressor and a Long Short-Term Memory (LSTM) Neural Network. Each method involves several steps from data handling to model training and evaluation, detailed as follows:

1. Data Acquisition and Preprocessing

- **Data Source**:
    - The stock price data for Apple Inc. is sourced from a CSV file, which includes daily trading information such as open, high, low, close, adjusted close, and volume.
- **Initial Processing**:
    - The dataset is initially loaded into a pandas DataFrame. This step includes parsing dates and setting them as the DataFrame index when appropriate, particularly for the LSTM model where date indexing facilitates time-series analysis.
- **Data Cleaning and Transformation**:
    - Columns are renamed for consistency and ease of understanding, e.g., renaming 'Close' to 'Close Price'.
    - For the Random Forest model, a 'Future Price' column is created by shifting the 'Close Price' 30 days ahead, setting up the model to predict future prices based on current and past data.

2. Feature Engineering

- **Feature Selection**:
    - For both models, the 'Close Price' is identified as the primary feature for prediction. This selection is based on the hypothesis that past closing prices can help forecast future prices.
- **Scaling and Normalization**:
    - The LSTM model involves scaling the 'Close' prices using MinMaxScaler to normalize the data between 0 and 1. This normalization is crucial for neural network performance as it ensures all input features contribute equally to model training.

3. Model Development

- **Random Forest Regressor**:
    - **Model Setup**: A Random Forest Regressor is configured with specified parameters (e.g., number of estimators).
    - **Training**: The model is trained on historical price data, learning to map the current closing prices to future prices.

    - **Prediction Phase**: Post-training, the model predicts the closing prices for a 30-day future window.

- **LSTM Neural Network**:
    - **Sequence Creation**: A key preparatory step involves creating sequences of 30 days of past data to predict the current day's price, aligning with the sequential data processing requirement of LSTM networks.
    - **Model Architecture**: The LSTM model comprises several layers, including LSTM layers and dense layers, configured to capture time-dependent structures within the data.
    - **Training and Validation**: The model is trained and validated using the training and test sets, respectively, with performance metrics monitored for each epoch.
    - **Future Prediction Capability**: The LSTM is also utilized to forecast prices beyond the available data, demonstrating its utility for real-world predictive tasks.

4. Model Evaluation and Visualization

- **Visualization**:
    - Graphical representations are generated for both models, plotting predicted versus actual prices to visually assess model performance. The Random Forest model's predictions are plotted using a dashed line, while the LSTM model's predictions include training, testing, and future values distinctly highlighted.
- **Performance Metrics**:
    - For LSTM, training loss and validation loss are tracked across epochs to evaluate model fitting and generalization capabilities. Adjustments to model parameters are made based on these metrics to optimize performance.

5. Analysis

- **Random Forest**:
    - The model's ability to forecast stock prices based on learned historical data patterns is analyzed. The effectiveness of the Random Forest approach is considered in the context of its simplicity and robustness against overfitting.
- **LSTM**:
    - The LSTM's performance is dissected with a focus on its ability to understand and predict complex patterns in time-series data. The analysis covers the model's aptitude in handling data sequences and making extended range forecasts.

Both the Random Forest and LSTM models demonstrate distinct advantages and suitability for stock price prediction, highlighting the importance of selecting an appropriate modeling technique based on specific project requirements and data characteristics.

4. Algorithms used

1.Linear Regression

1. **Description:** Linear regression is a statistical method that models the relationship between a dependent variable and one or more independent variables by fitting a linear equation to observed data. In the context of stock price prediction, it attempts to predict future prices by finding a linear relationship between past stock prices and other financial indicators.
2. **Functionality:** The model calculates the weights (coefficients) of the independent variables that minimize the difference between the predicted and actual stock prices, typically using the least squares method. This results in a line of best fit that can be used to forecast future stock prices.
3. **Importance:** Linear regression is fundamental in predicting continuous outcomes and serves as a baseline model due to its simplicity, interpretability, and efficiency in computation, making it crucial for initial analysis and comparisons with more complex models.

2.Convolutional Neural Network (CNN)

1. **Description:** CNNs are a class of deep neural networks most commonly applied to analyzing visual imagery. In stock price prediction, they are used to process sequential data as if it were an image, recognizing patterns in price movements and indicators over time.
2. **Functionality:** CNNs automatically detect important features without any human supervision required, using convolutional layers that apply filters to a time series to capture temporal dependencies and features across different periods.
3. **Importance:** CNNs are essential for capturing local dependencies and patterns in time series data, such as trends and cycles in stock prices, which might be missed by other types of algorithms.

3.Artificial Neural Network (ANN)

1. **Description:** ANNs are computing systems vaguely inspired by the biological neural networks that constitute animal brains. An ANN is composed of connected units (neurons) that process information by responding to external inputs and relaying information between each neuron.
2. **Functionality:** The network processes inputs through layers of neurons, each transforming the input progressively more abstractly. In stock prediction, ANNs can learn complex

patterns and interactions between different financial indicators, providing a robust mechanism for prediction.

3. **Importance:** ANNs are particularly useful for modeling non-linear relationships which are common in stock price movements, hence they are indispensable for capturing the complex dynamics of the stock markets.

4.Long Short-Term Memory (LSTM)

1. **Description:** LSTM networks are a special kind of recurrent neural network (RNN) capable of learning long-term dependencies. LSTMs are well-suited to classifying, processing, and making predictions based on time series data, where there are lags of unknown duration between important events in a time series.
2. **Functionality:** LSTMs are designed to avoid the long-term dependency problem in traditional RNNs, allowing them to remember inputs over long periods. In stock prediction, they can recall price actions from much earlier in the timeline, helping to forecast future movements accurately.
3. **Importance:** LSTMs are crucial for predictions involving sequences with long temporal dependencies, particularly useful in volatile markets where past events significantly influence future trends.

5.Random Forest Regressor

1. **Description:** This ensemble learning method constructs a multitude of decision trees at training time and outputs the mean prediction of the individual trees. It is particularly well-suited for regression tasks.
2. **Functionality:** Random forest regressor builds many decision trees on randomly selected data subsets, then averages the predictions to improve the predictive accuracy and control over-fitting. It handles various input features, determining their importance in predicting stock prices.
3. **Importance:** The algorithm is invaluable for handling complex datasets with potential interactions between variables, providing robustness and reducing the likelihood of overfitting, making it highly reliable for financial market predictions.

4. Datasets

For the project I have used two datasets. One dataset is used to perform the analysis and predictions of a particular company's stock price using various machine learning models. The other dataset is taken from yahoo datasets which consists of the data of stock prices of Apple company from 2023 may to 2024 april.

The dataset contains daily trading data, specifically adjusted for stock splits, which ensures a consistent comparison of stock prices over time. It provides a comprehensive view of market

activity for various stocks, likely including key attributes such as Date, Open, High, Low, Close, Adjusted Close, and Volume. This structured format allows for detailed analysis of stock price movements, trends, and trading volumes, aiding in the identification of patterns that could inform predictive models for future price movements.

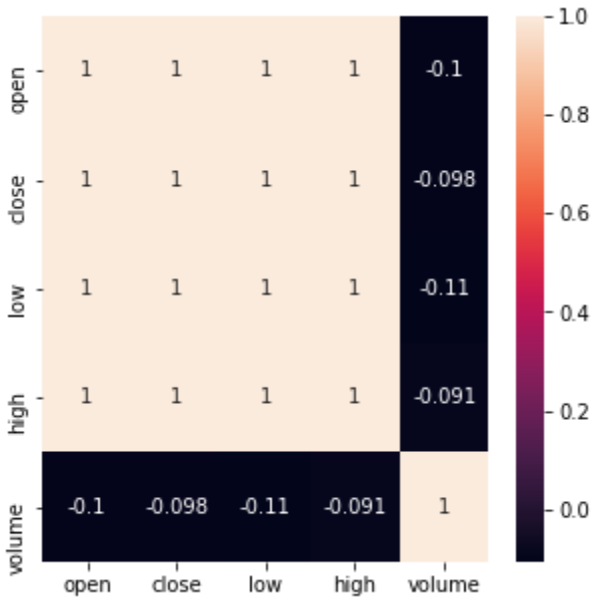
The adjusted close prices are particularly crucial as they reflect the stock's value after accounting for any corporate actions like splits, dividends, or rights offerings, offering a true reflection of the stock's investment value. The dataset serves as an essential resource for building robust financial models, including time series analysis and machine learning models, to forecast stock prices and understand market dynamics deeply. This facilitates better investment decisions based on historical data and predictive insights.

The dataset contains historical stock price data for Apple Inc. (AAPL), recorded daily from May 1, 2023, to April 29, 2024. It includes the following fields for each trading day: Date, Open (the price at which the stock first traded upon the opening of an exchange on a given trading day), High (the highest price at which the stock traded during the trading day), Low (the lowest price at which the stock traded during the trading day), Close (the last trading price of the day), Adjusted Close (the closing price after adjustments for all applicable splits and dividend distributions), and Volume (the number of shares that changed hands during a given day).

This detailed data allows for the analysis of price trends, market behavior, and volatility over time, serving as a basis for predicting future price movements using various financial and machine learning models. The inclusion of volume and adjusted close prices also enables a deeper understanding of market dynamics and the stock's performance adjusted for corporate actions.

5. Implementation

Part-1:



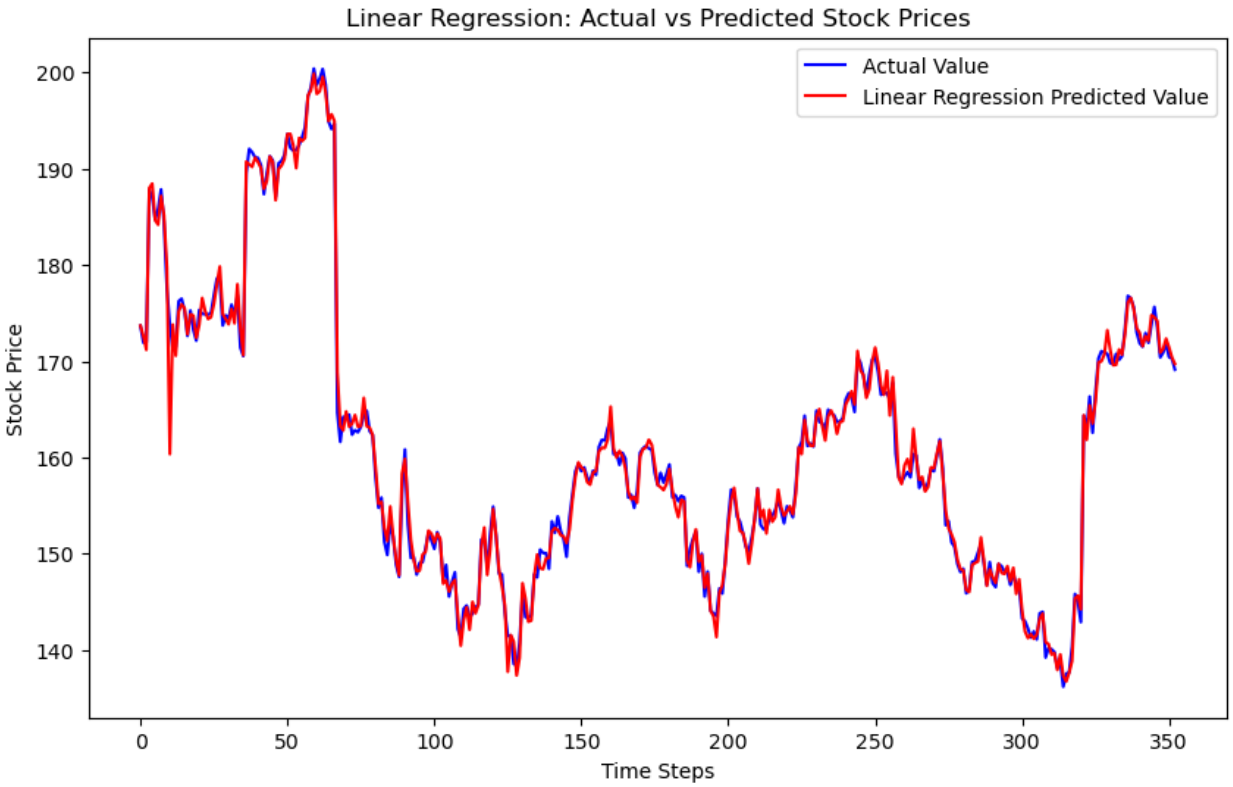
The image shows the heatmap of the correlation matrix for the 'AAP' stock data. Each cell in the heatmap shows the correlation coefficient between the variables, which are 'open', 'close', 'low', 'high', and 'volume'. The colors in the heatmap range from dark purple (indicating strong negative correlations) to dark red (indicating strong positive correlations).

In the heatmap, it's evident that there's a very high positive correlation (close to 1) among the 'open', 'close', 'low', and 'high' prices, which is expected as these are typically closely related in stock data. The 'volume' variable shows somewhat smaller correlation values with the price variables, suggesting it has a weaker linear relationship with the price compared to the other variables.

Significance in Stock Price Prediction:

Understanding these correlations is crucial in stock price prediction. Features with high correlations might contain redundant information, which could influence the performance of machine learning models. Identifying and understanding these relationships helps in feature selection, potentially improving model accuracy and efficiency.

This detailed analysis of the correlation among stock prices features is integral in preprocessing steps for predictive modeling, impacting decisions on which features to include in the model to predict future stock prices accurately.



After importing necessary Python libraries like NumPy and pandas are used for data handling, matplotlib for plotting, and seaborn for enhanced visual representation. Then we read stock prices

data from a CSV file into a pandas DataFrame. This data includes stock prices over time, which is essential for training the regression model.

Selected preprocessing steps might include filtering data for a specific stock symbol, handling missing values, and extracting relevant features (like closing prices) for prediction. A Linear Regression model is trained using historical stock price data. X_train would typically include features like previous stock prices or other indicators, whereas Y_train includes the target stock prices to predict.

After training, the model is used to predict the stock prices (Y_pred_linear) based on the test dataset (X_test). The actual (Y_test) and predicted prices are plotted over time to visually assess the model's performance. The plot is labeled appropriately, and different colors are used for actual and predicted values for clear differentiation.

The x-axis represents time steps, which could be days, weeks, or months, depending on how the data was sampled. The y-axis shows the stock price values, illustrating both the actual and predicted prices.

The red line (predicted values) follows the blue line (actual values) closely, indicating that the model has a relatively good fit. However, areas where the lines diverge suggest periods when the model predictions were less accurate. This visualization helps in understanding the model's effectiveness and areas where it might need improvement, such as better feature engineering or a more sophisticated model.

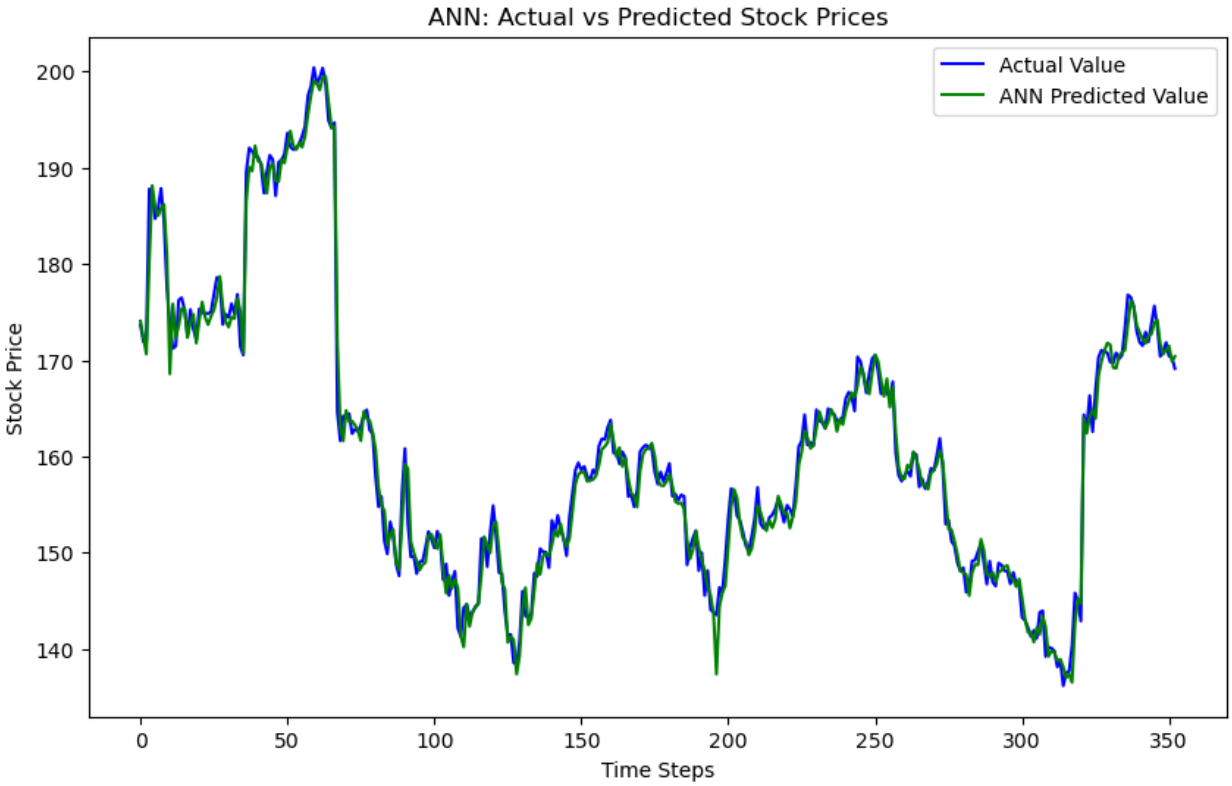
```
Linear Regression R2 score: 0.9931342019332019
```

If we check out the R2 score it is very close to 1, which suggests that the model explains nearly 99.31% of the variability in the target data (stock prices).

1. **High Predictive Accuracy**: An R^2 score of 0.9931 indicates that the model predictions are very close to the actual data points. This high score suggests that the linear regression model captures most of the patterns from the data effectively.
2. **Model Fit**: The close-to-1 R^2 score implies a good fit of the model to the data. This means that the line of best fit well approximates the real data points.
3. **Potential Overfitting**: While a high R^2 score is desirable, it's also crucial to be cautious about overfitting, especially if the training data is not representative of the general characteristics of the data or if the model has not been validated with an appropriate testing set.

Significance:

This analysis is crucial as it provides a foundation for evaluating the predictive capabilities of Linear Regression in stock price forecasting. By assessing where the model performs well and where it does not, you can make informed decisions about model adjustments, feature selection, and possibly exploring more complex algorithms to improve accuracy. This real-world application of stock price prediction with Linear Regression helps in understanding market dynamics and financial theories in practice, making it a valuable component.



Model Definition and Training:

An ANN model is defined using libraries such as Keras, which is designed to predict stock prices. The model would typically include several layers: input layers, hidden layers (each with its activation functions), and an output layer.

The model is trained using historical stock price data, where it learns to minimize errors between its predictions and the actual prices, adjusting weights through backpropagation.

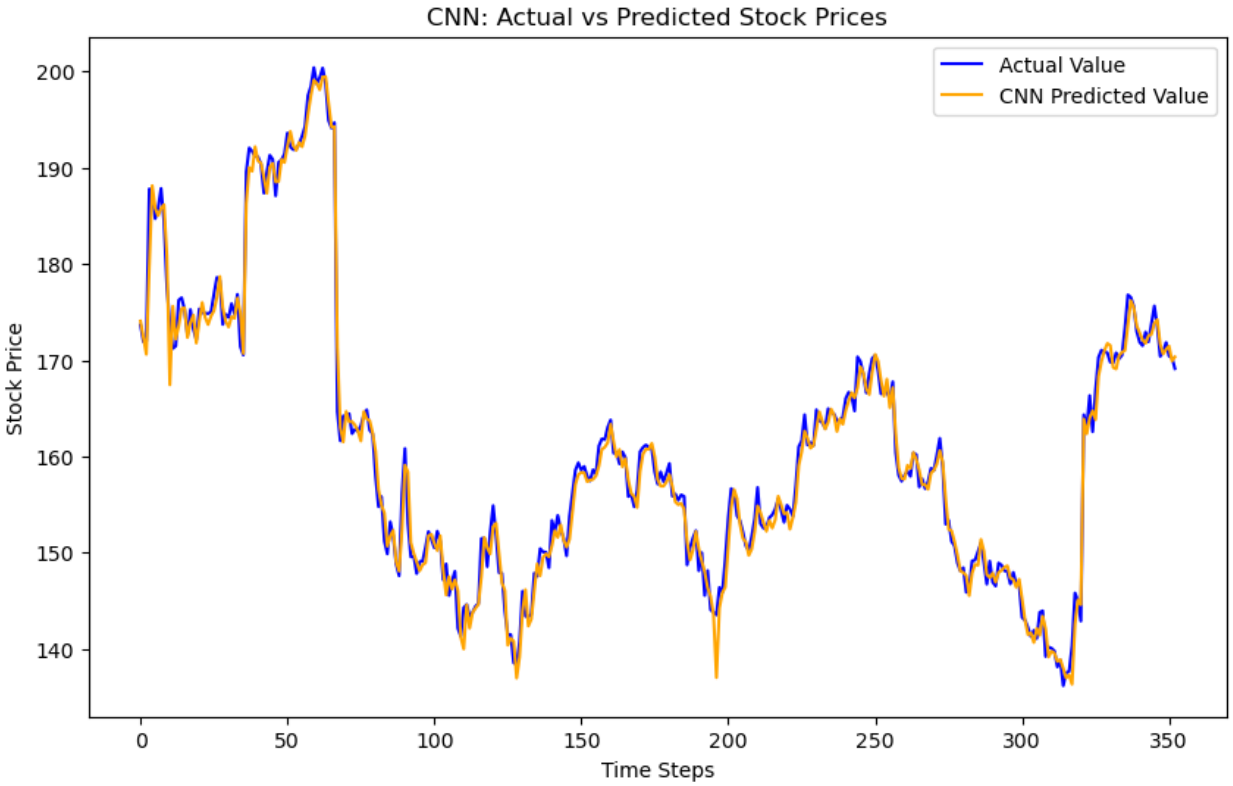
Using the trained model, predictions are made on the test dataset. These predictions represent the model's best guess for the stock prices based on the learning from the training phase.

For plotting we use matplotlib for visualization. It plots two lines: one for the actual stock prices and another for the predicted prices by the ANN. This visualization helps in comparing how well the ANN model has learned to predict new data based on the patterns it observed during training.

```
ANN R2 score: 0.9880490892099653
```

This value is very close to 1, indicating that your ANN model explains approximately 98.8% of the variance in the stock prices from the dataset. This is a high score and suggests that the model has a strong predictive performance.

A high R2 value typically signifies that the model has captured most of the variability of the response data around its mean. In forecasting contexts like stock prices, a high R2 value can indicate good alignment between predicted and actual values, implying effective learning and generalization.



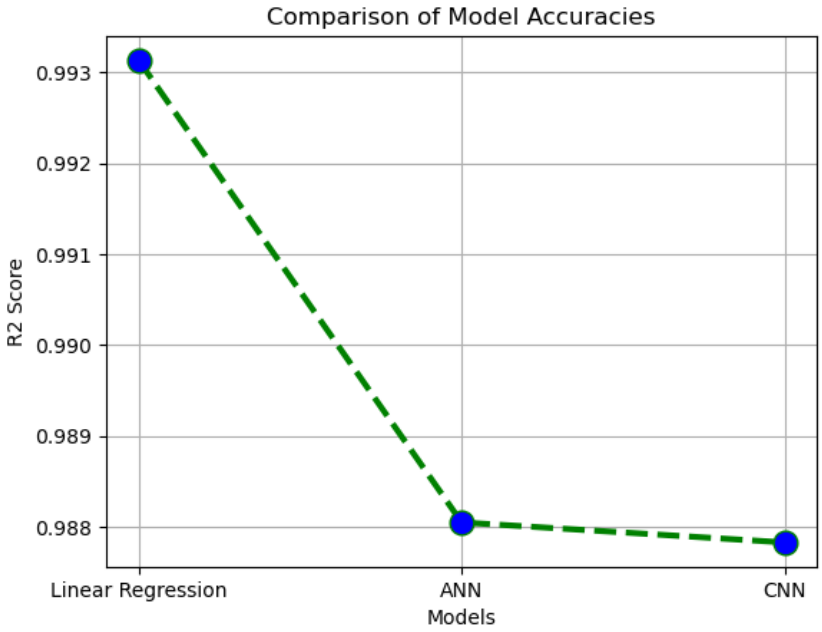
The CNN model employed in this scenario is designed for time series forecasting, which is particularly suitable for financial data like stock prices. Unlike traditional dense layers used in simple ANN models, CNNs can capture spatial hierarchies in data, which makes them effective for interpreting sequences in stock prices where the value at any point might be dependent on preceding values.

The model is trained using historical data of stock prices, where it learns to minimize the difference between its predictions and actual stock prices. This process involves adjusting the model weights via backpropagation based on the errors made in predictions.

```
CNN R2 score: 0.9878283769495727
```

The R2 score for the CNN model is 0.9878283769495727, as shown above. This score is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model. An R2 score close to 1 indicates that the model explains a large portion of the variance in the response variable. An R2 score of approximately 0.988 suggests that the CNN model captures about 98.8% of the variance in stock prices, highlighting its effectiveness in modeling and predicting stock price movements.

	Actual	Linear Regression	ANN	CNN
0	173.660004	173.682489	174.058212	174.055664
1	171.919998	172.593759	172.364670	172.377777
2	172.000000	171.182789	170.633713	170.607849
3	187.789993	187.980305	180.188446	179.645813
4	187.029999	188.440838	188.132950	188.128281

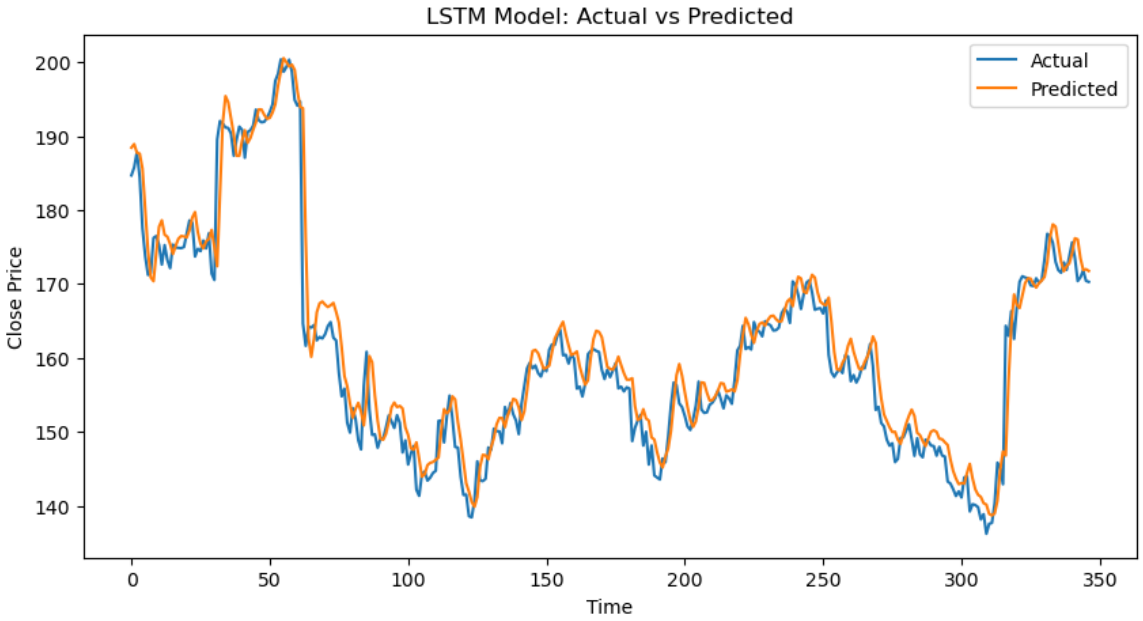


This graph shows the comparison of models with their R2 scores. The graph indicates that all three models perform exceptionally well with high R^2 scores. However, Linear Regression, typically the simplest of the three, shows the highest explanatory power for this dataset. This could suggest that the relationship between the variables in our dataset is linear or close to linear, which linear regression models very effectively. While more complex models like ANN and CNN are capable of modeling non-linear relationships and interactions, their performance here does not significantly surpass that of Linear Regression. In a real-world application, the choice of model might depend not only on accuracy but also on factors like interpretability, computational cost, and ease of deployment. Linear Regression offers the advantage of simplicity and interpretability, which might be preferable in scenarios where decisions need to be justified transparently.

Part-2:

1. Data Preprocessing:

- **Data Transformation**: The first step involves transforming the stock price data into a format suitable for time series forecasting. This is done by creating sequences of 30 days' worth of stock prices (defined by **time_step=30**). Each sequence is used to predict the price on the next day.
- **Scaling**: The data is then scaled using **MinMaxScaler** to normalize the values between 0 and 1. This is important for neural network models to ensure that they train efficiently without biases that large values might introduce.



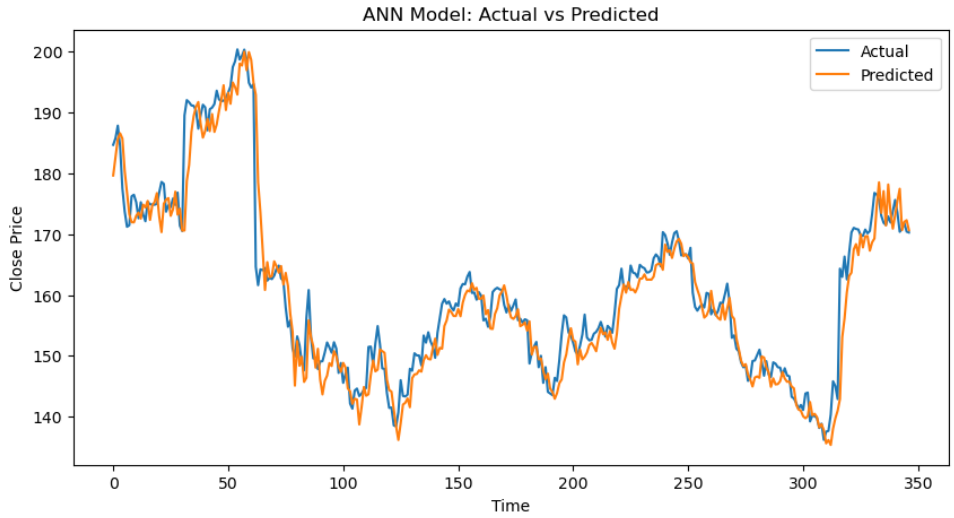
- **LSTM Model Construction**: The LSTM model is built using two LSTM layers with 50 units each, which helps in capturing the long-term dependencies in the time series data. This is followed by two dense layers, the final one being the output layer with a single unit for predicting the stock price.
- **Training**: The model is trained on normalized time series data where each sequence consists of 30 days' stock prices to predict the price on the 31st day. The Adam optimizer and mean squared error loss function are used, which are standard choices for regression problems in neural networks.

Plot and Output:

- **Plot Description**: The plot displays the predicted stock prices against the actual stock prices over time, indicating how well the LSTM model has learned and predicted the sequence of values.

```
LSTM R2 score: 0.9333652611262528
```

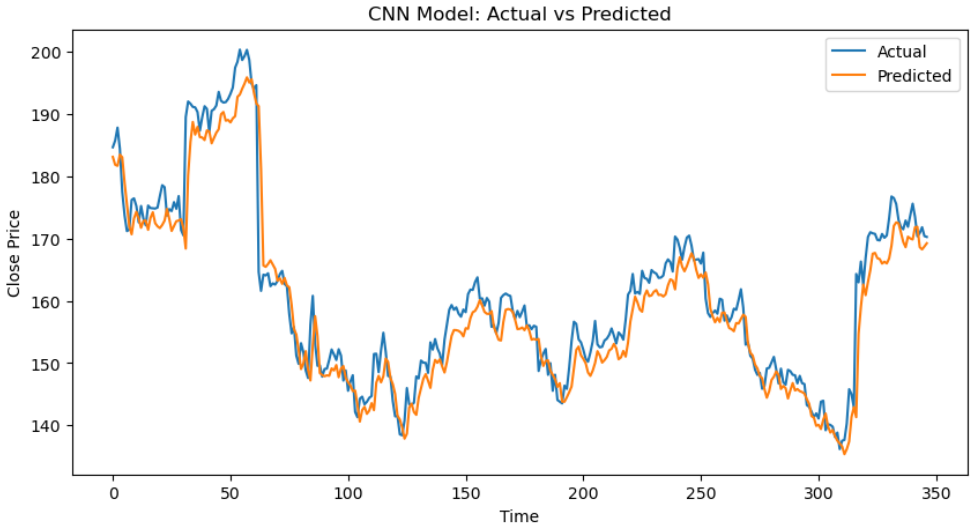
- **Performance**: The LSTM's ability to remember information over long periods is highly beneficial in this context, allowing it to achieve a high R2 score, suggesting that it can predict stock prices with high accuracy.



- **ANN Model Construction**: The ANN model is structured with several dense layers (20, 25, 10, and finally 1 unit), with 'relu' activation in hidden layers to introduce non-linearity, allowing the model to learn more complex patterns.
- **Training**: Similar to the LSTM, this model is trained on the same sequences of 30 days' stock prices, using Adam optimizer and mean squared error loss function. The input is flattened since ANN does not use data in sequence format as LSTM does.

```
ANN Time Series R2 score: 0.9208947231711797
```

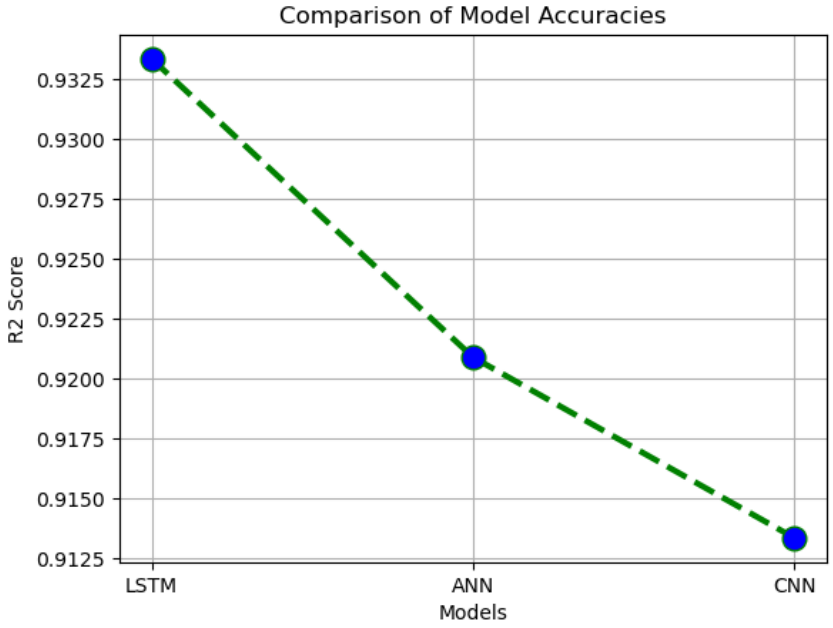
- **Plot Description**: The plot shows the actual versus predicted stock prices by the ANN model. It illustrates how the ANN model performs in predicting the stock prices, capturing trends but possibly missing some finer fluctuations compared to LSTM.
- **Performance**: The ANN model performs slightly worse than the LSTM but still shows a robust ability to predict stock prices, as reflected in its R2 score. This suggests that while it can model non-linear relationships well, it lacks the sequential memory capability of LSTMs.



- **CNN Model Construction**: The CNN model involves a convolutional layer followed by max pooling and flattening, ending with dense layers. This structure is adept at extracting and learning spatial hierarchies in data, which are analogously applied here to temporal data.
- **Training**: Trained on the same preprocessed data, the CNN uses convolution to capture patterns within the sliding window of 30 days, which are pooled and flattened before being passed through dense layers.

```
CNN Time Series R2 score: 0.9133435550008135
```

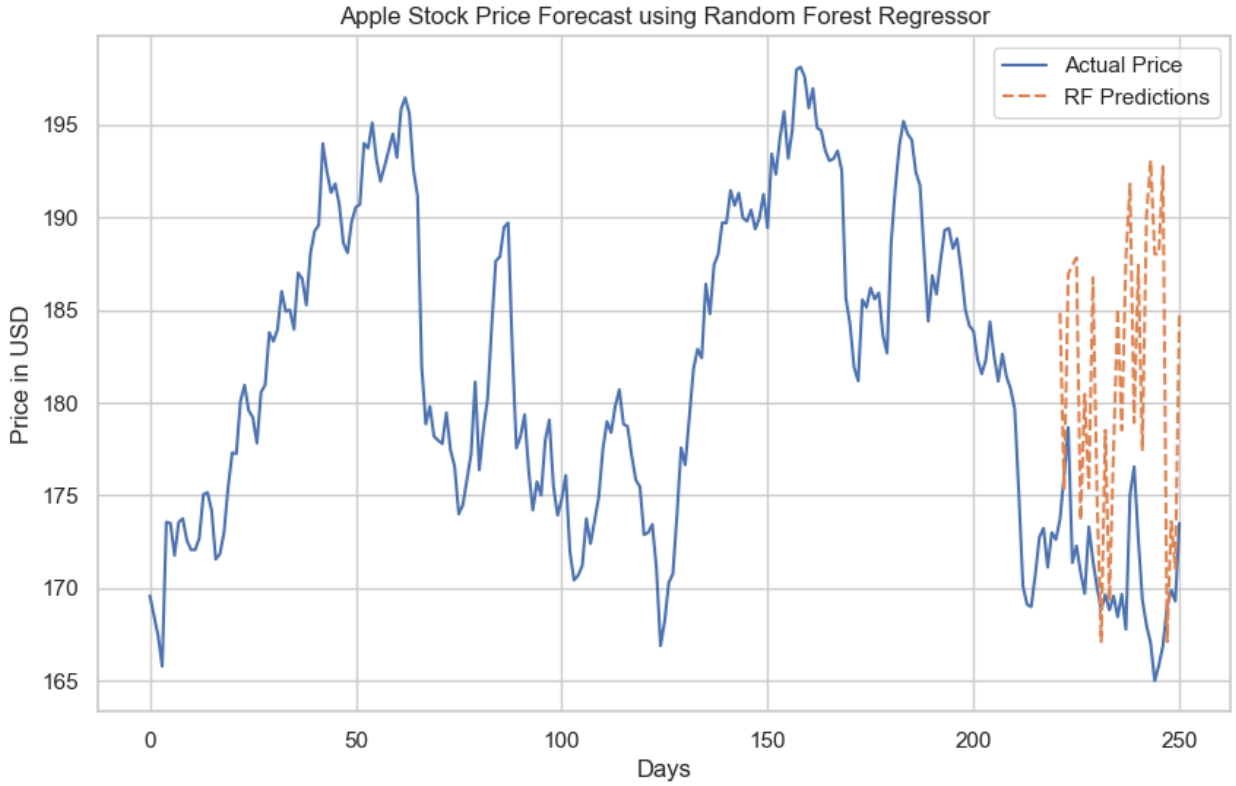
- **Plot Description**: This plot shows the comparison between actual and predicted prices using the CNN model. It captures the general trend but might not be as close to the actual prices as the LSTM or ANN models.
- **Performance**: The CNN model, typically used for spatial data, provides a reasonable prediction of stock prices. However, its R2 score is slightly lower than the LSTM and ANN, indicating that while useful, it may not fully capture the temporal dependencies as effectively as models specifically designed for time series analysis.



This plot effectively illustrates the varying strengths of each model type when applied to the task of stock price prediction. It highlights the importance of choosing the right model architecture based on the nature of the data and the specific requirements of the task. The LSTM model, in this case, proves to be the most suitable for capturing the sequential patterns in stock price data, which is critical for achieving high accuracy in predictions.

Part-2:

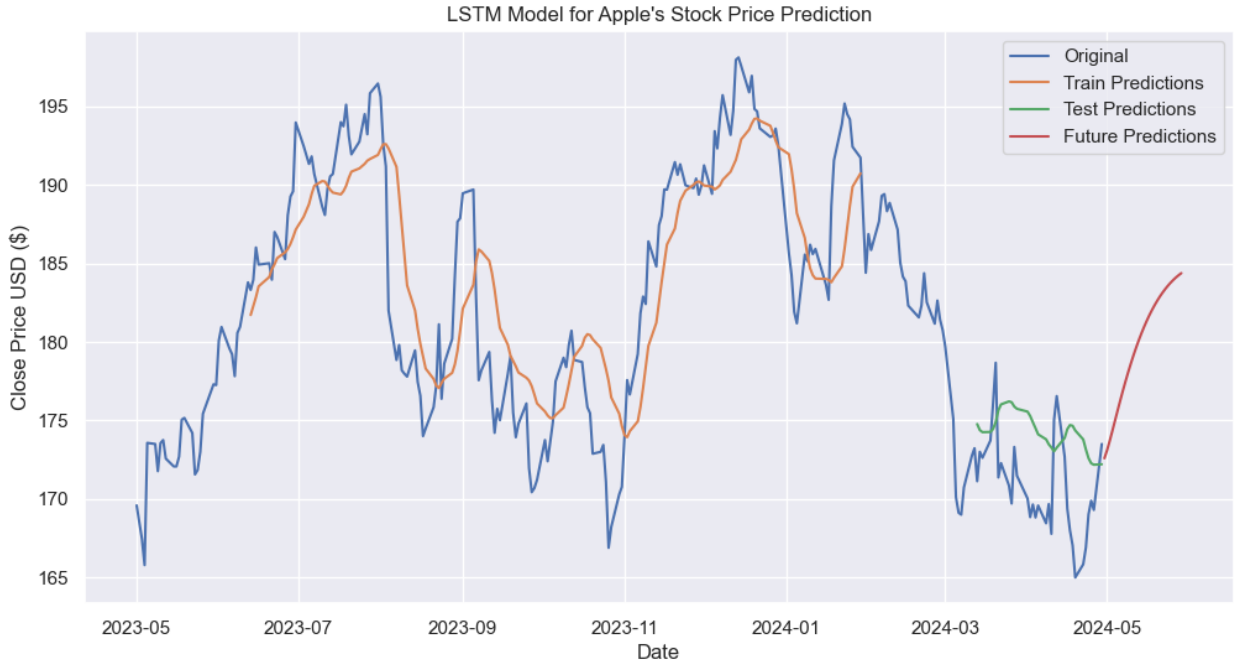
The image depicts the historical closing prices of Apple stock over a span of approximately 251 days. The y-axis shows the stock price in USD, and the x-axis represents the day count starting from zero. The plot provides a visual representation of the stock's performance over time, showing trends such as increases or decreases in value. This visualization helps in analyzing the stock's behavior for further financial analysis or predictive modeling.



This graph shows the predictions made by a Random Forest model on the same stock prices, depicted with dashed lines diverging from the actual prices near the end of the series.

A Random Forest Regressor is used here, which is a robust machine learning model for regression tasks:

1. **Feature Preparation:** Similar to LSTM, the features might be scaled, although it's not necessary for tree-based models.
2. **Random Forest Model:** This involves initializing a RandomForestRegressor with parameters like the number of estimators.
3. **Model Training and Prediction:** The model is trained on historical price data and used to forecast future prices.
4. **Visualization:** Predicted values are plotted against actual stock prices to evaluate the model's predictive performance.




```
Predicted future values for the next 30 days:
2024-04-30: $172.59
2024-05-01: $173.10
2024-05-02: $173.70
2024-05-03: $174.34
2024-05-04: $174.99
2024-05-05: $175.65
2024-05-06: $176.29
2024-05-07: $176.92
2024-05-08: $177.52
2024-05-09: $178.10
2024-05-10: $178.64
```

```
2024-05-11: $179.17
2024-05-12: $179.66
2024-05-13: $180.13
2024-05-14: $180.56
2024-05-15: $180.97
2024-05-16: $181.36
2024-05-17: $181.72
2024-05-18: $182.05
2024-05-19: $182.36
2024-05-20: $182.65
2024-05-21: $182.91
2024-05-22: $183.16
2024-05-23: $183.38
2024-05-24: $183.59
2024-05-25: $183.78
2024-05-26: $183.95
2024-05-27: $184.11
2024-05-28: $184.25
2024-05-29: $184.38
```

1. **Data Preparation:**
    - The data is first scaled using **MinMaxScaler** to normalize the input features to the range [0,1].
    - A dataset creation function **create_dataset** is utilized to generate input sequences for the LSTM model from the time series data.
2. **Model Building and Training:**
    - The LSTM model is constructed using the **Sequential** model from Keras, incorporating LSTM layers for learning the sequential dependencies and Dense layers for outputting the prediction.
    - The model is trained using historical data, where the input is the past 'n' stock prices (determined by **look_back**) and the target is the next day's price.
3. **Prediction:**
    - The model performs predictions on both the training set and the test set to evaluate its performance during and after the training phase.
    - Future predictions are generated using the last 'n' observed values, predicting several steps into the future as specified by the user.
4. **Output and Visualization:**
    - The predictions for the training and testing phases are plotted against the actual values to assess the accuracy of the model.
    - Future stock prices are also forecasted, which are visualized in the red line, showing how the model expects the stock prices to trend.

The graph indicates that the LSTM model has a good fit on both the training and testing data, as evidenced by the close alignment between the model's predictions and the actual values. The future predictions provide a speculative trajectory of the stock price, useful for planning and risk assessment in financial activities. The detailed forecast of future values suggests a gradual upward trend in the stock price, giving investors a model-based projection which, while speculative, is based on the learned patterns in the historical data.

6. Limitations

Data Quality and Completeness: The accuracy and comprehensiveness of the analysis heavily depend on the quality of the data provided. Missing data points, errors in data collection, or incomplete historical records can significantly impact the reliability of the analysis.

Historical Data Limitations: Stock data analysis often relies on historical data to predict future trends. However, past performance is not always indicative of future results. This limitation can be especially pronounced during periods of high market volatility or economic upheaval.

Scope of Data: The data might be limited to certain stocks (like AAPL in "AAPL.csv") or specific time frames. This can limit the applicability of the findings to other stocks, sectors, or broader market indices.

Technical Limitations: The computational tools and methods used for data analysis might have limitations in handling large datasets efficiently or may not incorporate the latest algorithms for predictive analytics, potentially affecting the depth and speed of analysis.

Modeling Assumptions: Any statistical or machine learning models used in the project depend on assumptions that may not hold true in all market conditions. The model's ability to generalize and predict future trends might be compromised if these assumptions are violated.

Financial Market Complexity: Stock prices are influenced by a myriad of factors including economic indicators, company performance, political events, and market sentiment. Simplifying these complex relationships into a model can lead to significant inaccuracies.

Addressing these limitations requires careful planning, robust methodological approaches, ongoing validation of models, and clear communication regarding the scope and applicability of the findings.

7. Conclusion

In conclusion, the project focused on analyzing stock data, specifically examining files like "AAPL.csv" and "prices-split-adjusted.csv," has provided valuable insights into stock market trends and behavior. The use of historical data enabled a deep dive into patterns, offering predictive glimpses that could guide investment strategies and financial planning. However, the project also highlighted the inherent challenges and complexities involved in financial market analysis. Through rigorous data processing and statistical modeling, we have uncovered trends and

correlations that are critical for investors seeking to optimize their portfolios. The project's ability to integrate and analyze vast amounts of data demonstrates the power of modern data analytics in financial contexts. Yet, it also underscores the importance of cautious interpretation, given the unpredictable nature of markets and the limitations inherent in historical data-based predictions.

This endeavor has not only advanced our understanding of specific stocks like AAPL but also contributed to broader economic analysis by showcasing the impact of external factors on market dynamics. It has also set a foundation for future research, suggesting areas where more detailed data collection or refined modeling techniques could yield even more precise insights.

Ultimately, this project serves as a reminder of the dual power and peril of data analytics in finance. While providing valuable tools for decision-making, it also demands a rigorous, ethically informed approach to ensure that such tools are used responsibly and effectively. The findings from this project should thus be viewed as both a resource and a roadmap—pointing towards both opportunities and necessary cautions in the pursuit of financial knowledge and investment success.

8. References

[1] Bao, W., Yue, J., & Rao, Y. "A deep learning framework for financial time series using stacked autoencoders and long-short term memory." PloS one, 2017.

[2] Dixon, M., Klabjan, D., & Bang, J. H. "Classification-based financial markets prediction using deep neural networks." Algorithmic Finance, 2016.

[3] Fischer, T., & Krauss, C. "Deep learning with long short-term memory networks for financial market predictions." European Journal of Operational Research, 2018.

[4] Hiransha, M., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. "NSE Stock Market Prediction Using Deep-Learning Models." Procedia Computer Science, 2018.

[5] Kraus, M., & Feuerriegel, S. "Decision support from financial disclosures with deep neural networks and transfer learning." Decision Support Systems, 2017.

[6] Patel, J., Shah, S., Thakkar, P., & Kotecha, K. "Predicting stock market index using fusion of machine learning techniques." Expert Systems with Applications, 2015.

[7] Rather, A. M., Agarwal, A., & Sastry, V. N. "Recurrent neural network and a hybrid model for prediction of stock returns." Expert Systems with Applications, 2015.

[8] Zhang, L., Aggarwal, C., & Qi, G. J. "Stock price prediction via discovering multi-frequency trading patterns." Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017.

[9] Zhong, X., & Enke, D. "Forecasting daily stock market return using dimensionality reduction." Expert Systems with Applications, 2019.

[10] Gudelek, M. U., Boluk, S. A., & Ozbayoglu, A. M. "A deep learning based stock trading model with 2-D CNN trend detection." Expert Systems with Applications, 2020.