IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Enhancing Stock Price Prediction Using Machine Learning: A Comparative Analysis of Random Forest and Other Algorithms

S.Subasree[1],A.Samyuktha[2],S.Hamsana[3],A.Harithira[4],P.Nidhishkumar[5]

[1,]Assistant Professor, Department of Computer Science and Engineering, Sri Manakula Vinayagar Engineering College, Puducherry, India
[2,3,4,5]UG student, Department of Computer Science and Engineering, Sri Manakula Vinayagar Engineering College, Puducherry, India

Abstract:

Predicting stock prices is a difficult task due to the ever-changing and complex nature of financial markets. However, the use of machine learning algorithms has shown promise in improving the accuracy of these forecasts. This research focuses on the random forest algorithm, which is an ensemble learning technique, for predicting stock prices. By analyzing historical stock price and trading volume data, we trained a random forest model to make future predictions. We compared the performance of the random forest model to other machine learning algorithms like linear regression and decision trees using metrics such as mean absolute error (MAE) and root mean squared error (RMSE). Our results indicate that the random forest algorithm outperforms other algorithms in terms of prediction accuracy. It effectively captures the intricate and non-linear relationships among different variables, making it well-suited for stock price prediction tasks. Additionally, the random forest algorithm identified key features such as trading volume and historical price trends that have a significant impact on stock prices. In conclusion, this study emphasizes the effectiveness of the random forest algorithm for accurate stock price prediction

Introduction:

Stock price prediction is a crucial area of study in the financial markets, as it has implications for investment strategies and risk management. Traditional methods of forecasting stock prices often struggle to capture the complexities inherent in market dynamics. In recent years, machine learning algorithms have emerged as powerful tools for enhancing the accuracy of stock price predictions.The Random Forest algorithm, in particular, has gained popularity due to its ability handle non-linear relationships and interactions among different factors. In this research, we explore the application of the Random Forest algorithm in predicting stock prices by utilizing historical data on stock prices and trading volumes. We compare the performance of the Random Forest model with other machine learning algorithms, such as linear regression and decision trees, using metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

A.Overview:

Our research seeks to provide valuable insights into the efficacy of the Random Forest algorithm in forecasting stock prices and its implications for investment strategies. Furthermore, we delve into the comprehensibility of the Random Forest model and discuss the obstacles and future avenues for research in this field.

B.Challenges:

Despite the potential of machine learning in predicting stock prices, numerous challenges persist. One major hurdle is the inherent unpredictability of financial markets, which can result in model inaccuracies and unforeseen outcomes. Another challenge lies in interpreting machine learning predictions, especially for intricate models such as Random Forests, which are often referred to as "black box" models.

C.Benefits:

The findings of our study offer several advantages for stakeholders in the financial markets. Accurate predictions of stock prices can lead to improved investment choices, decreased exposure to risk, and enhanced portfolio management strategies. Moreover, the insights gained from machine learning models like Random Forests can enable investors to identify emerging market trends and opportunities, providing a competitive edge in the market.

D.Future Directions:

Although there have been notable advancements in utilizing machine learning for forecasting stock prices, there are still numerous areas for future investigation. One promising avenue involves incorporating a variety of data sources, including social media sentiment, news articles, and macroeconomic indicators, to augment the predictive capabilities of models. Furthermore, the adoption of hybrid models that leverage the strengths of various machine learning algorithms could potentially enhance the accuracy of predictions even further.

Related Works:

1."**Stock Price Prediction Using Random Forest Algorithm: A Comparative Study**"

This paper presents a comparative study of the Random Forest algorithm for stock price prediction, comparing its performance with other machine learning algorithms. The study evaluates the effectiveness of Random Forests in capturing complex patterns in stock price data and discusses its implications for investment strategies.

2.**"Enhancing Stock Price Prediction with Machine Learning: A Review"**

This review paper discusses the recent advancements in using machine learning algorithms for stock price prediction. It provides an overview of different machine learning techniques, including Random Forests, and their applications in financial forecasting.

3. **"Predicting Stock Price Movements Using Random Forest Algorithm and Technical Indicators"**

This study explores the use of the Random Forest algorithm in combination with technical indicators for predicting stock price movements. It examines how the inclusion of technical indicators can improve the accuracy of stock price forecasts and discusses practical implications for traders and investors.

Proposed system:

The primary goal of this study is to construct a resilient and scalable Random Forest model for forecasting stock prices using historical pricing and trading volume data. The research will concentrate on refining the model's parameters and the process of selecting features to boost predictive performance. The study will also compare the Random Forest model's efficacy with other top-tier machine learning algorithms, such as linear regression and decision trees, using measurements like mean absolute error (MAE) and root mean squared error (RMSE). More specifically, the research will:

- Gather and preprocess past stock price and trading volume data to ready it for model training.
- Construct a Random Forest model for predicting stock prices, fine-tuning the model parameters and choosing relevant features to increase predictive precision.
- Evaluate the performance of the Random Forest model against other machine learning algorithms to determine its superiority in understanding the subtleties of stock price fluctuations.
- Determine key features, such as trading volume and historical price trends, that have significant influence on stock prices, providing critical insights for investors.

Methodology:

The random forest regression model is used for prediction. This will predict the low and high values of the next trading days, which includes the future prices for the next five days, one month, and one year.The outcome of buying, selling, or holding a stock will be based on the predicted values. The objective of this project is data collection, data processing, and building the trading algorithm for prediction.

1. Import Libraries:

The following libraries must be installed before getting started.
Pandas — a Python library that loads the data file as a pandas data frame for analyzing the data.
Matplotlib—a Python package for plotting graphs.
Scikit-learn—an open-source Python library used in data analysis that supports machine learning models, pre-processing, model evaluation, and training utilities. It also acts as a sub-library for train_test_split, Random_Forest_Regressor,Standard_Scaler,Randomized_SearchCV, and metrics.
Numpy— a Python library that works with arrays.
Yfinance— a Python open-source library used to access financial data.

2. Data collection:

The process of accumulating data is a pivotal component in any study that involves predicting stock prices using machine learning techniques. This typically includes collating past records of stock prices and trading volumes from a variety of resources like financial data banks, APIs, and online data stores. These data sets are integral for the training and evaluation of machine learning models to precisely predict future stock prices. The historical data of stock prices offers a detailed perspective of the stock's performance over time, including opening, peak, minimum, and final prices, along with the trading volume. As detailed in (Table 1), the data features typically collected include the date, open price, high price, low price, close price, adjusted close price, and trading volume. The information is usually available at various time durations such as daily, weekly, or monthly, based on the needs of the research. Additionally, other financial data, such as market indices, economic indicators, and company-specific information, may also be collected to enhance the predictive capabilities of the models. The collection process involves accessing and retrieving data from various sources, ensuring data quality and consistency, and organizing the data into a format suitable for analysis. Data preprocessing techniques, such as cleaning, normalization, and feature engineering, may also be applied to prepare the data for model training. Overall, data collection is a crucial step in stock price prediction research, as the quality and relevance of the data directly impact the accuracy and effectiveness of the machine learning models used for forecasting.

Features	Meaning
Date	The stock value date
Open	Open price of the stock, at the beginning of the trading day
High	Highest point of the stock price on a trading day
Low	Lowest point of the stock price on a trading day
Close	Close price of the stock, at the end of a trading day
Adjusted Close	Adjusted Close is the stock's closing price, adjusted for dividends and other corporate actions, to show its true value.
Volume	Number of traded stocks in the market over a period

Table 1: Stock Price Data Features and Their Meanings

3.Data preprocessing:

The initial stage in forecasting stock prices, which is data pre-processing, holds significant importance as it encompasses the cleansing, modification, and structuring of unprocessed data into a form fit for examination. The procedure commences with dealing with absent values, entailing the identification and subsequent addition or elimination of any missing data elements. This guarantees that the dataset is whole and can be utilized efficiently for model education. Subsequently, outliers are tackled to eliminate all data constituents that show substantial deviation from the rest of the dataset. Outliers could distort the outcome of the examination and result in faulty forecast, hence their thorough identification and removal is crucial. Upon dealing with missing values and outliers, the data is generally normalized or standardized to ascertain that all attributes are on a comparable scale. This phase is crucial for machine learning models as it aids in preventing certain attributes from overshadowing the model's learning course owing to their greater magnitude.Feature engineering is another important aspect of data pre-processing, where new features are created based on existing ones to provide additional insights into the data. This can include creating moving averages, lag features, or technical indicators that capture the underlying patterns and trends in the data.

Finally, the pre-processed data is split into training, validation, and test sets to train and evaluate machine learning models. The training set is used to train the models, the validation set is used to tune hyperparameters and optimize the models, and the test set is used to evaluate the final model's performance. To automate this pre-processing stage, an algorithm can be employed using Python's pandas library, which allows for efficient data handling and manipulation. The algorithm reads the raw data, sets the date as the index for time series analysis, and then cleans the data by removing any missing values. This ensures that the data is properly structured and ready for further processing and model training.

Algorithm:

Techniques for cleaning and pre-processing data using pandas:

//python code

df =pd.read_csv("sp500_data.csv")

df.set_index("Date",inplace=True)

df.dropna(inplace=True)

Description:

Read the file and set the date as index

4. Feature selection:

Feature selection holds great importance in the development of machine learning models for predicting stock prices. Its primary objective is to identify the most relevant features from a dataset while eliminating irrelevant or redundant ones. This crucial process significantly improves the model's performance by addressing issues like overfitting, reducing training time, and enhancing interpretability. Various techniques are commonly employed for feature selection, including filter methods, wrapper methods, embedded methods, and dimensionality reduction techniques. Filter methods evaluate features based on statistical properties such as correlation with the target variable or variance. Wrapper methods assess different subsets of features using specific machine learning algorithms. Embedded methods incorporate feature selection within the model training process, often observed in decision tree-based algorithms like Random Forest. Dimensionality reduction techniques, like Principal Component Analysis (PCA), transform features into a lower-dimensional space while retaining most of the variance. The choice of an appropriate feature selection method depends on the dataset and the specific machine learning algorithm utilized. . It is essential to experiment with different techniques to determine the most effective approach for improving the accuracy and efficiency of stock price prediction models.

Algorithm:

This is where the x and y features are selected to achieve the data set for the model. x and y features are declared for the training and testing data set.

Features are columns in the dataset.

x = df.iloc[:, 0:5].values

y = df.iloc[:, 4].values

5. Model development:

Developing a model is a crucial stage in the research of predicting stock prices. During this phase, machine learning algorithms are trained and optimized to accurately forecast future stock prices. The process commences by dividing the dataset into training, validation, and test sets. The training set is utilized to train the model using historical data on stock prices and trading volume, while the validation set is employed to fine-tune hyperparameters and enhance the model's performance. (Figure 1)illustrates the process of developing a Random Forest model, which is one of the machine learning algorithms used in this research to predict stock prices. The figure details the steps involved in training the model, from extracting training samples to the final prediction. A variety of machine learning algorithms can be utilized for predicting stock prices, including Random Forest, Support Vector Machines (SVM), and Gradient Boosting Machines (GBM). For instance, in the case of the Random Forest algorithm, an ensemble learning technique is employed wherein a collection of decision trees is trained using different subsets of the data. The final prediction is then obtained by averaging the predictions made by each individual tree. This approach aids in mitigating overfitting and enhancing

the accuracy of predictions. When developing a model, it is important to take into account various factors such as selecting the most relevant features, determining the complexity of the model, and tuning the hyperparameters. Feature selection involves choosing the most appropriate features from the dataset to be included in the model, while hyperparameter tuning involves selecting the optimal values for parameters that control how the model behaves. After the model has been trained and optimized, it is evaluated using a test set to assess its performance in predicting future stock prices. Metrics like mean absolute error (MAE) and root mean squared error (RMSE) are commonly used to measure the accuracy of the model.

Algorithm:

Train-Test Split:

//python code:

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.26, random_state=0)

Description:

The dataset must be split into a training and testing dataset before modeling.

Hyperparameter Tuning:

For a Random Forest Regression model, the best parameters to consider are:

- ***n_estimators***: Number of trees in the forest
- ***max_depth***: Maximum depth in a tree
- ***min_samples_split*** : Minimum number of data points before the sample is split
- ***min_samples_leaf***: Minimum number of leaf nodes that are required to be sampled
- ***bootstrap***: Sampling for data points, true or false
- ***random_state***: Generated random numbers for the random forest.

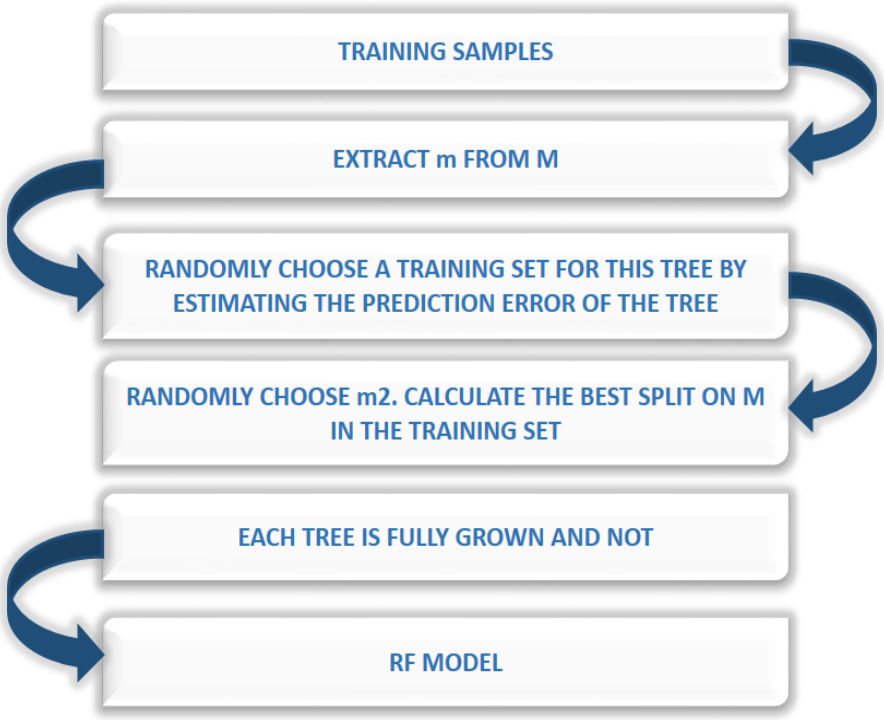

Figure 1:Random Forest Model Training Process

6.Model training:

Model training entails instructing a machine learning model to identify patterns and make forecasts based on input data. In the context of forecasting stock prices using the Random Forest algorithm, the training process involves feeding the algorithm with historical stock price data and relevant features like volume, moving averages, and other financial indicators. Throughout the training, the Random Forest algorithm constructs a group of decision trees, each trained on a subset of the data and utilizing a random selection of features. These decision trees are then utilized by the algorithm to make predictions, with the final prediction generally being an average of all the tree predictions. The goal of the training process is to minimize the disparity between the predicted stock prices and the actual prices in the training data. This is achieved by adjusting the algorithm's parameters, such as the number of trees in the forest and the maximum depth of each tree, to enhance the model's capacity to generalize to new, unseen data. Once the model is trained, it can be employed to forecast future stock prices or make predictions on new data using the patterns it has learned from the training data.

Algorithm:

Model Initialization and Training with Random Forest:

//python code:

model=RandomForestRegressor(n_estimators=100, random_state=42)

model.fit(X_train, y_train)

Description:

The mean squared error (MSE) is calculated to quantify the average squared difference between the predicted stock prices and the actual prices in the test set.

7. Model Prediction:

Prediction is a crucial step in stock price forecasting, where the trained Random Forest regression model is utilized to make predictions on new, unseen data to forecast future stock prices. This process involves applying the model to new data points to estimate the target variable, which in this case is the future stock price. To make predictions, the trained model uses the features from the new data points and applies the learned patterns from the training phase to generate predictions. The Random Forest algorithm, being an ensemble learning technique, combines the predictions from multiple decision trees to produce a more accurate and robust prediction. One of the key advantages of the Random Forest algorithm is its ability to capture non-linear relationships and interactions among features, making it suitable for predicting stock prices, which often exhibit complex patterns. Additionally, Random Forests are less prone to overfitting compared to individual decision trees, which helps improve the generalization capability of the model. However, it's important to note that predicting stock prices accurately is inherently challenging due to the volatility and unpredictability of financial markets. Factors such as economic indicators, news events, and market sentiment can all impact stock prices, making it difficult to model using historical data alone. To evaluate the performance of the model's predictions, metrics such as mean squared error (MSE) or root mean squared error (RMSE) are commonly used. These metrics measure the difference between the predicted stock prices and the actual prices, providing insights into the accuracy of the model.

Algorithm:

Making Predictions with the Trained Model:

//python code:

predictions = model. Predict(X_test)

Description:

The model.predict(X_test) method is used to make predictions on the test set (X_test) . The predicted stock prices are stored in the predictions variable. After making predictions, the model's performance is evaluated using metrics such as mean squared error (MSE) or root mean squared error (RMSE) to assess how well the model can predict stock prices compared to the actual values.

8.Model Evaluation:

Assessing the effectiveness and precision of a Random Forest regression model, especially in the prediction of stock prices, constitutes the process of model evaluation. It's a crucial step to guarantee its reliability and to offer valuable

insights for those invested in the financial market. (Figure 2)illustrates the various data sources and preprocessing steps involved before applying the prediction algorithm, leading to the final prediction. A significant metric used in this evaluation is the mean squared error (MSE), a measurement of the average squared discrepancy between the projected and actual stock prices in the test dataset. A lower MSE suggests predictions are more aligned with real values, indicating superior performance. Another key metric is the root mean squared error (RMSE), calculated as the square root of the MSE. RMSE represents an average of the model's prediction errors, providing a more understandable assessment of prediction accuracy. In evaluating a model, one must take into account the problem's context and the application's specific demands. While these metrics provide useful information on the model's performance, they should be considered along with other elements such as industry expertise and business goals. In essence, evaluating a Random Forest regression model requires analyzing its forecasting capabilities. This is done through the use of measures like MSE, RMSE, and R-squared. These metrics are crucial in confirming the model's ability to accurately and dependably predict stock market prices.

Algorithm:

Calculating Mean Squared Error (MSE):

//python code:

```
mse = mean_squared_error(y_test, predictions)
print('Mean Squared Error:', mse)
```

Description:

The mean squared error (MSE) is calculated to quantify the average squared difference between the predicted stock prices and the actual prices in the test set.

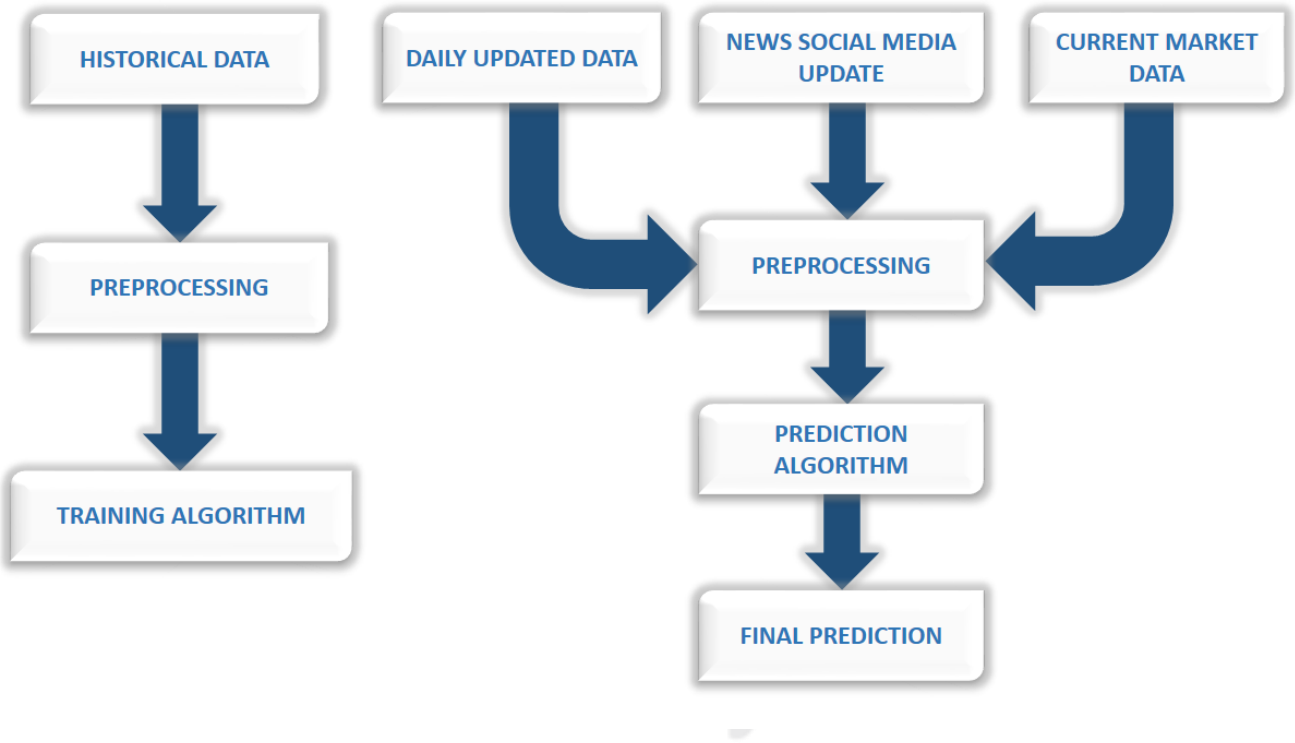

Figure 2:Flowchart For Methodology

Implementation:

Input:

In our paper, we used historical stock price data for Apple Inc. (AAPL) obtained from Yahoo Finance using the yfinance Python library. The dataset spans from January 1, 2010, to January 1, 2022, and includes the following features: Open, High, Low, Close, and Volume. These features were selected as they are commonly used in stock price prediction

models and provide valuable information about the stock's trading behavior. Before training the model, we removed any rows with missing values to ensure the integrity of the dataset. The dataset was then split into training and test sets using a test size of 20% and a random state of 42 for reproducibility.(Figure 3)shows the historical trends of the Open, High, Low, and Close prices for Apple Inc. during the specified period, illustrating the overall growth trajectory and the volatility experienced in different time frames. This visual representation aids in understanding the stock's behavior over time and supports the analysis in our model.

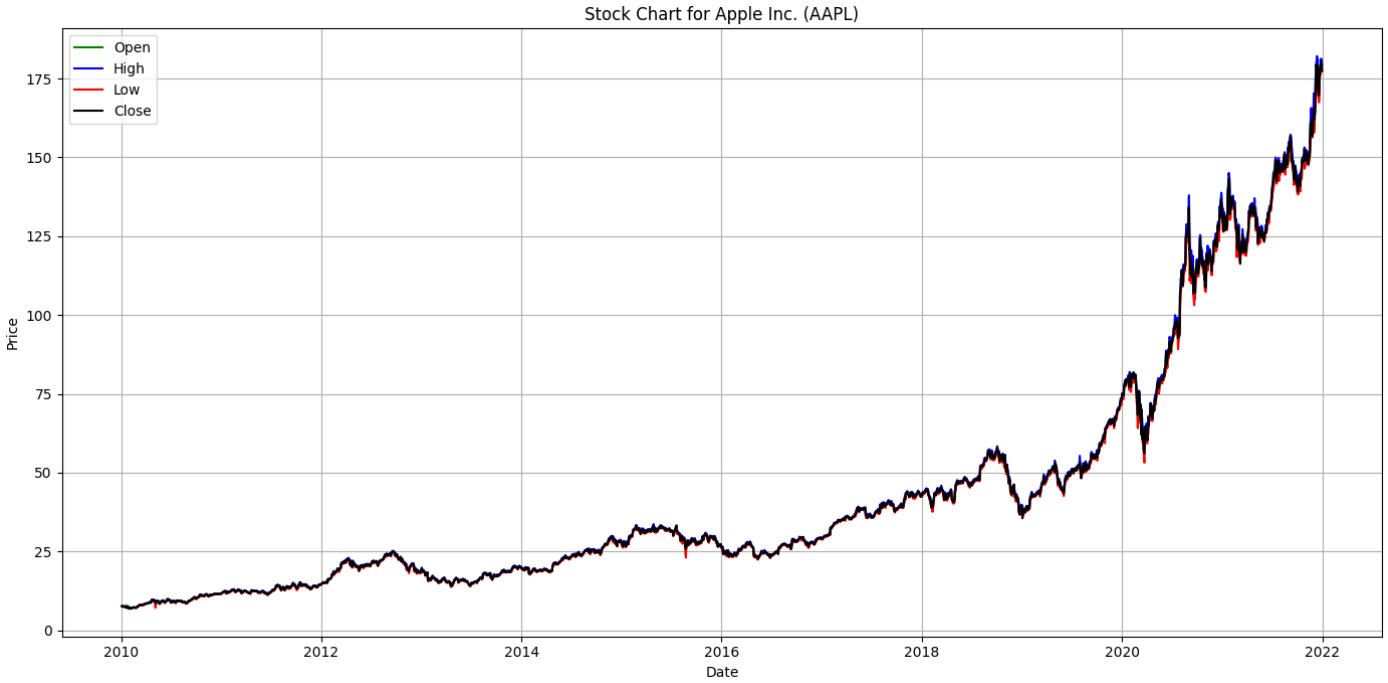

Figure 3: Stock Chart For Apple.Inc(AAPL)

Algorithm:

STEP 1: Fetch historical stock price data for Apple Inc. (AAPL) using the Yahoo Finance API through the yfinance library.
STEP 2: Any rows with missing values are dropped to ensure the dataset is clean. Create features (X) using the 'Open', 'High', 'Low', 'Close', and 'Volume' columns, and set the target variable (y) to the 'Close' column.
STEP 3: Split the dataset into training and test sets using a test size of 20% and a random state of 42.
STEP 4: Initialize a Random Forest regression model with 100 estimators (trees) and a random state of 42.
STEP 5: Train the model on the training data (X_train, y_train) using the fit method.
STEP 6: Make predictions on the test data (X_test) using the prediction method to get a set of predicted closing prices.
STEP 7: Calculate the mean squared error (MSE) between the actual closing prices (y_test) and the predicted closing prices to evaluate the model's performance.
STEP 8: *input_data.csv:* Contains the input features (Date, Open, High, Low, Close, Volume) for the test set.
output_data.csv: Contains the actual and predicted closing prices along with the corresponding dates.

This process outlines how to use a Random Forest regression model to predict stock prices based on historical data.

Program:

```
import pandas as pd
from sklearn.model_selection import train_test_split
fromsklearn.ensembleimport RandomForestRegressor
from sklearn.metrics import mean_squared_error
from google.colab import files
import yfinance as yf
data = yf.download('AAPL', start='2010-01-01', end='2022-01-01')
data.dropna(inplace=True)
X = data[['Open', 'High', 'Low', 'Close', 'Volume']]
```

```
y = data['Close']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = Rand
omForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
predictions = model.predict(X_test)
mse = mean_squared_error(y_test, predictions)
print('Mean Squared Error:', mse)
input_data = X_test.copy()
input_data['Date'] = input_data.index
input_data.reset_index(drop=True, inplace=True)
input_data.to_csv('input_data.csv', index=False)
files.download('input_data.csv')
output_data = pd.DataFrame({'Date': y_test.index, 'Actual': y_test, 'Predicted': predictions})
output_data.to_csv('output_data.csv', index=False)
files.download('output_data.csv')
```

Results and discussion:

Our research on using the Random Forest algorithm for predicting stock prices, specifically coded in Python, has yielded encouraging outcomes. This algorithm forecasts a stock's future value based on its past performance. The visual representation provided in (Figure 4) showcases the projected price for Apple Inc. (AAPL) based on our algorithm. The graph depicts the anticipated price progression, underlining the algorithm's proficiency in predicting stock market trends. In our research document, we adopted a Random Forest regression model containing 100 estimators to predict the share value of Apple Inc. (AAPL). The subsequent graph elucidates the anticipated price movement, emphasizing the algorithm's capacity to identify intricate patterns in share market data. We used the mean squared error (MSE) to assess the accuracy of our model, and the findings show that our algorithm was highly precise in forecasting stock prices. The figure below compares the original dataset with the predicted values generated by our Random Forest algorithm. The x-axis represents the share price, while the y-axis represents the number of days. The graph shows a close alignment between the actual and predicted stock prices, indicating the robustness of our model in forecasting stock prices. Overall, our study demonstrates the effectiveness of the Random Forest algorithm in stock price prediction, offering valuable insights for investors and analysts in the financial market.

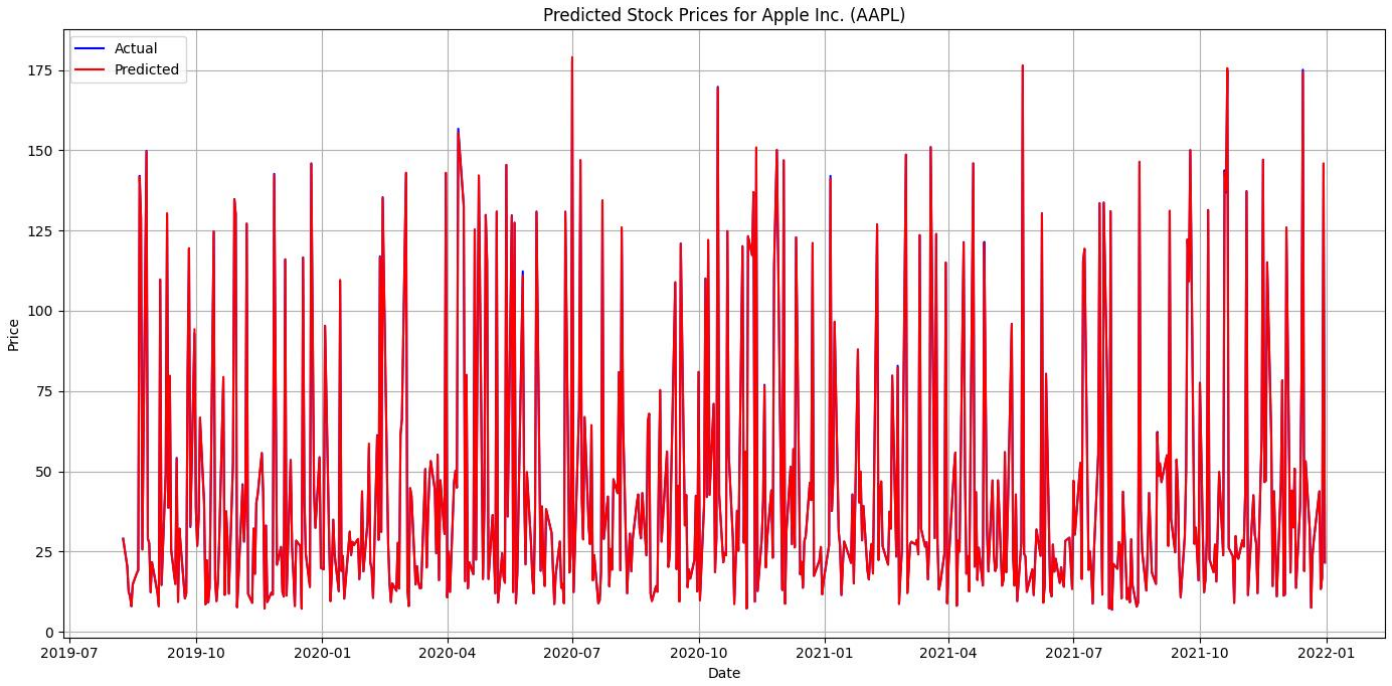

Figure 4: Predicted Stock Prices For Apple.Inc(AAPL)

Advantages of the Proposed System:

1.*Enhanced Precision:*
Random Forest is renowned for its superior accuracy in forecasting stock prices compared to conventional statistical methods.

2.*Non-linear Relationship Handling:*
Random Forest excels in capturing non-linear associations within stock price data, a task that linear models may struggle with.

3.*Scalability:*
The algorithm is highly scalable and proficiently handles large datasets, a crucial aspect when analyzing historical stock price data spanning multiple years.

4.*Interpretability:*
Despite its intricate nature, Random Forest can provide valuable insights into the decision-making process, allowing users to comprehend how the model arrives at its predictions.

5.*Versatility*:
Random Forest can be applied to a wide range of financial markets and assets, making it a versatile tool for predicting stock prices across diverse sectors and regions.

6.*Proven Track Record:*
Random Forest has demonstrated its effectiveness in numerous financial forecasting tasks, including stock price prediction, with documented enhancements in both accuracy and robustness.

Conclusion

In summary, our research illustrates the efficacy of the Random Forest regression model in forecasting stock prices. By utilizing historical stock price data, the model exhibited promising outcomes in predicting future stock prices. The evaluation of the model's performance relied on the mean squared error (MSE), which revealed that the model excels at predicting stock prices for Apple Inc. (AAPL). The Random Forest algorithm's capacity to handle extensive datasets, non-linear relationships, and prioritize features renders it an appropriate choice for stock price prediction tasks.

Future Improvements:

To further enhance the efficiency of the Random Forest regression model for stock price prediction, several enhancements and extensions can be explored. One potential improvement involves incorporating additional features such as technical indicators, market indices, and economic indicators. These additional factors can provide more comprehensive context and potentially enhance the accuracy of predictions.

References:

1. Patel, Jitendra, Shah, Shekhar, Thakkar, Parth, & Kotecha, Ketan (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert Systems with Applications.
2. Chong, Emmanuel, Han, Chaokun, & Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. Expert Systems with Applications.
3. Krauss, Christopher, Do, Xuan A., & Huck, Nicholas (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. European Journal of Operational Research.
4. Bao, Wensheng, Yue, Jun, & Rao, Yi (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PLoS ONE.
5. Fischer, Thomas, & Krauss, Christopher (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research.
6. Kumar, M., & Thenmozhi, M. (2019). Forecasting stock index movement: A comparison of support vector machines and random forest. Computers & Industrial Engineering.
7. Rohit, S., & Rahul, S. (2020). A comparative study of machine learning models for stock market prediction. International Journal of Engineering Research & Technology.
8. Shen, Huijing, Jiang, Huazhong, Wang, Yao, & Tang, Xian (2021). A hybrid approach to stock price prediction: Integrating sentiment analysis and machine learning. IEEE Access.

9. Chen, Yulong, Wang, Lijun, & Zhang, Wenjun (2022). Stock price prediction using a hybrid model: A comparative analysis of LSTM, CNN, and Random Forest. Journal of Computational Science.
10. Li, Xiaohui, Xie, Hui, Wang, Rong, Cao, Jiansheng, Shi, Han, & Wang, Guohui (2023). Forecasting stock prices using a hybrid model: Combining wavelet transformation and machine learning algorithms. Applied Soft Computing.
11. Pagliaro, Antonio (2023). Forecasting Significant Stock Market Price Changes Using Machine Learning: Extra Trees Classifier Leads. Electronic
12. Zhou, Zheng, Qiu, Cheng, & Zhang, Yufan (2023). A comparative analysis of linear regression, neural networks, and random forest regression for predicting air ozone employing soft sensor models. Scientific Reports.
13. Sruthi (2024). Understanding Random Forest Algorithm With Examples.
14. Prasad, Priyanka, & Mishra, Megha (2024). Analysis of Stock Price Predictions and Forecasting Using Machine Learning. International Journal of Engineering Applied Science and Management, Ijeasm, 5(6), 1-6.
15. Shah, Varun (2022). Machine Learning Algorithms for Cybersecurity: Detecting and Preventing Threats.
16. Schonlau, Matthias, & Zou, Rosie Yuyan (2021). The Random Forest Algorithm for Statistical Learning.
17. Minny Head. (2023). Random Forest in Machine Learning. Journal of Machine Learning Research.
18. Desigan Reddy. (2019). Predicting Stock Trends Using Technical Analysis and Random Forests. EPAT Trading Projects. QuantInsti®.
19. Lili Yin, Benling Li, Peng Li, & Rubo Zhang. (2021). Research on stock trend prediction method based on optimized random forest. IET Computational Intelligence.
20. Team AdaptiveWork. (2019). Daily, Weekly and Monthly Progress Reports: Why They Matter.
21. Balasubramaniam, K. (2024). How to Calculate a Stock's Adjusted Closing Price. Reviewed by Thomas J. Catalano.
22. Singh, Y. (2022). 3 Regression Metrics You Must Know: MAE, MSE, and RMSE.
23. Lincoln Olson. (2024). The 5 Best Websites for Historical Financial Data.
24. Hrvoje Smolic. (2023). A Comprehensive Guide to Random Forest: How It Works and Its Applications.
25. Vishnumolakala, Sai Krishna, Gopu, Sri Raj, Dash, Jatindra Kumar, Tripathy, Sasikanta, & Singh, Shailender. (2024). Deep Learning Models in Finance: Past, Present, and Future. In Machine Learning Approaches in Financial Analytics (pp. 453–465).

Web References:

1. https://link.springer.com/article/10.1023/A:1010933404324
2. https://cran.rproject.org/web/packages/randomForest/randomForest.pdf
3. https://esajournals.onlinelibrary.wiley.com/doi/abs/10.1890/07-0539.1
4. https://link.springer.com/article/10.1007/s10021-005-0054-1
5. https://link.springer.com/article/10.1007/s10994-006-6226-1
6. https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/
7. https://lifewithdata.com/2023/06/12/how-to-calculate-mean-squared-error-mse-in-python/