|  |
| --- |
| **Stock market price prediction**  **A Machine Learning Approach** |

**PROJECT SUBMITTED TO ASIAN SCHOOL OF MEDIA STUDIES**

**IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE**

**AWARD OF**

**DIPLOMA**

**in**

**Data Science**

By

**Suraj Kumar Pandit**

**Under the Supervision of**

**Prof. Manpreet Kaur Bhatia**

****

ASIAN SCHOOL OF MEDIA STUDIES

SCHOOL OF DATA SCIENCE

**2024**

**DECLARATION**

**I, Suraj kumar pandit, S/O Krishna pandit,** declare that my project entitled

**Stock market price prediction : A machine Learning Approach** submitted at **School of Data science, Asian School of Media Studies, Film City, Noida, for the award of Diploma in Data Science, ASMS** is an original work and no similar work has been done in India anywhere else to the best of my knowledge and belief.

This project has not been previously submitted for any other degree of this or any other University/Institute.

** *Signature***

**Suraj kumar pandit**

**6202843547**

**Surajkumar.hzb63@gmail.com**

**Diploma in Data Science**

**School of Data Science**

**Asian School of Media Studies**

**ACKNOWLEDGEMENT**

The completion of the project titled **Stock Market price prediction : A machine Learning Approach**, gives me an opportunity to convey my gratitude to all those who helped to complete this project successfully. I express special thanks:

* **To *Prof. Sandeep Marwah,* President,** Asian School of Media Studies, who has been a source of perpetual inspiration throughout this project.
* To ***Mr. Ashish Garg,*** Director **for** School of Data Science for your valuable guidance, support, consistent encouragement, advice and timely suggestions.
* To ***Mrs. Manpreet Kaur Bhatia***,Assistant Professor of School of Data Science, for your encouragement and support. I deeply value your guidance.
* To my ***all faculty*** & ***friends*** for their insightful comments on early drafts and for being my worst critic. You are all the light that shows me the way.

To all the people who have directly or indirectly contributed to the writing of this report, but their names have not been mentioned here.

***Signature***

**Suraj Kumar Pandit**

**6202843547**

**Surajkumar.hzb63@gmail.com**

**Diploma in Data Science**

**School of Data Science**

**Asian School of Media Studies**

**ABSTRACT**

Among investors, the ability to predict future developments or crises in the stock market has long been highly valued. During the COVID-19 worldwide pandemic, this skill became even more crucial, underscoring the need of risk management in preserving stability in such unpredictable times.

Although there is a rising need for reliable intelligent systems that can precisely anticipate stock prices to inform investment strategies, traditional business research still uses a variety of risk management techniques. Nowadays, a large portion of this field's research focuses on applying machine learning techniques to predict stock price patterns. While these approaches have shown encouraging results, there aren't many thorough surveys that list all of the machine learning algorithms used for stock price prediction.

Stock Market is one of the most vibrant sectors in the financial system, marking an important contribution to economic development. Stock Market is a place where buyers and sellers of securities can enter into transactions to purchase and sell shares, bonds, debentures etc. In other words Stock Market is a plate form for trading various securities and derivatives. Further, it performs an important role of enabling corporate, entrepreneurs to raise resources for their companies and business ventures through public issues. Today long term investors are interested to invest in the Stock market rather than invest anywhere. The Bombay Stock Exchange (BSE), the National Stock Exchange (NSE) and the Calcutta Stock Exchange (CSE) are the three large stock exchanges of Indian Stock Market.

The main objective of present study is to present review of literature related to Indian Stock Market to study the Indian Stock Market in depth. The study would facilitate the reader to know the past, current and future trend or prospects of Indian Stock market. This study would provide guidelines to investor to maximise profit with minimize risks. High degree of volatility in the recent times in the Indian market has led to more development in the future.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
|  | ***Page No.*** | |
| **Declaration** | | 1 |
| **Acknowledgment** | | 2 |
| **Abstract** | | 3 |
| **List of figures** | | 7 |
| **CHAPTER 1: Introduction**  1.1 Introduction 9  1.1.1 Background 9  1.1.2 Problem Statement 10  1.1.3 Objectives 10  1.1.4 Outline of the study 10-12    2. Literature Review 12- 14  3. Definitions 16  **CHAPTER 2: Dataset Preparation/Pre-processing**   * 1. Introduction 16   2. Exploratory Data Analysis 18-24   **CHAPTER 3: Model Selection: Algorithms of ML**  Model selection 24-26  3.1 Random forest 26-27   * + 1. Mathematical Intuition 27-21     2. Implementation with the dataset 28-44   **CHAPTER 4: Analysis of Result & Discussion**   * 1. Experimental work 45-48      1. Performance Measures/Evaluation Metrics 48-53   **CHAPTER 5: Conclusion**   * 1. Summary 54-55   2. Future Scope of Work 55   **REFERENCES 56** | |

**LIST OF FIGURES**

Fig 1 Plot adjusted close over time

Fig 2 EDA

Fig 3 Plot adjusted close over time

Fig 4 Histogram plot

Fig 5 KDE plot

Fig 6 Fluctuation plot

Fig 7 30 days moving average

Fig 8 100 days moving average plot

Fig 9 Comparision between close price and predicted Price

Fig 10 Compare last 15 days and 30 days graph

Fig 11 plotting whole closing price with prediction

Fig 12 comparision between original closing price and prediction price

# RELIANCE INDUSTRIES

### STOCK ANALYSIS AND FORECASTING

## Business Objective

Predict the Reliance Industries Stock Price for the next 30 days.

There are Open, High, Low and Close prices that you need to obtain from the web for each day starting from 2015 to 2022 for Reliance Industries stock.

* Split the last year into a test set- to build a model to predict stock price.
* Find short term, & long term trends.
* Understand how it is impacted from external factors or any big external events.
* Forecast for next 30 days.

## Collection of Dataset

* For this project, we will be using the Yfinance library to get the data, which makes it easy to process.
* We collected data from 1-Jan-2015 to 28-Feb-2023.
* But also you can download data from ‘Yahoo! Finance’ website. You can use Below link.
* <https://finance.yahoo.com/quote/RELIANCE.NS/history?p=RELIANCE.NS>

## About the data

* Date: Date of trade
* Open: Opening Price of Stock
* High: Highest price of stock on that day
* Low: Lowest price of stock on that day
* Close: Close price adjusted for splits.
* Adj Close: Adjusted close price adjusted for splits and dividend and/or capital gain distributions.
* Volume: Volume of stock on that day

**CHAPTER 1**

**Stock Market Price Prediction**

* 1. **INTRODUCTION**

"Stock Market Price Prediction using Random Forest" aims to leverage statistical techniques to predict stock market movements. By employing Random Forest, this project seeks to identify and quantify the relationships between various factors and stock prices. The analysis will use historical data to model and forecast future stock trends, providing insights into market behaviour and aiding investment decisions. This method of forecasting is grounded in finding the linear correlation between variables, allowing for a simplified yet effective approach to understanding and predicting market fluctuations.

### 1.1.1 BACKGROUND AND MOTIVATION

The stock market is a complex, dynamic, and influential component of the global economy, reflecting the collective actions and expectations of investors worldwide. Predicting stock prices has always been a significant challenge due to the market's inherent volatility and the multitude of factors influencing price movements. Traditional methods of stock price prediction, such as fundamental and technical analysis, rely heavily on historical data and financial ratios. However, these methods often fall short in capturing the intricate patterns and dependencies within the data.

With the advent of machine learning, more sophisticated techniques have emerged, offering improved accuracy and insights. Among these, the Random Forest algorithm stands out for its robustness, flexibility, and ability to handle large datasets with high dimensionality. Random Forest, an ensemble learning method, combines multiple decision trees to enhance predictive performance and reduce overfitting, making it particularly suitable for stock price prediction.

.

**1.1.2 PROBLEM STATEMENT**

This project aims to leverage the power of Random Forest to predict the closing prices of Reliance Industries Limited's stock. By analyzing historical stock data, the project seeks to develop a model that can provide accurate predictions, aiding investors in making informed decisions. The primary challenge lies in handling the volatile nature of stock prices and ensuring the model generalizes well to new, unseen data.

**1.1.3 OBJECTIVES**

The main objectives of this project are:

* To preprocess and analyze the historical stock price data of Reliance Industries Limited.
* To engineer relevant features that can enhance the model's predictive capability.
* To build and optimize a Random Forest model for accurate stock price prediction.
* To evaluate the model's performance using appropriate metrics and validate its reliability.

### 1.1.4 SIGNIFICANCE OF STUDY

Accurate stock price prediction is crucial for investors, financial analysts, and policymakers. It can lead to better investment strategies, risk management, and economic stability. By utilizing Random Forest, this study aims to contribute to the field of financial analytics, demonstrating the potential of machine learning in solving complex predictive tasks. The insights gained from this project can help investors make better decisions, ultimately leading to improved financial outcomes.

### 1.1.5 DEFINITIONS

* **Random Forest:** A machine learning algorithm that builds multiple decision trees and merges them to get a more accurate and stable prediction.
* **Stock Price Prediction:** The process of forecasting the future price of a company's stock based on historical data and other influencing factors.
* **Machine Learning:** A subset of artificial intelligence that involves training algorithms to make predictions or decisions without being explicitly programmed to perform the task.

### 1.1.6 OVERVIEW

This project involves collecting and preprocessing the dataset, engineering features, building a Random Forest model, and evaluating its performance. The findings and insights from the model's predictions will be discussed, providing a comprehensive understanding of the model's effectiveness and areas for improvement.

**1.1.4 OUTLINE OF STUDY**

 Introduction

* Overview of stock market prediction
* Importance of accurate stock market price prediction
* Introduction to Random Forest as a predictive tool

 Literature Review

* Historical approaches to stock market prediction
* Advances in machine learning and their application in stock prediction
* Key findings from previous studies using Random Forest in stock market analysis.

 Research Objectives

* To identify the key factors affecting stock prices
* To develop a Random Forest model for stock price prediction
* To evaluate the accuracy and effectiveness of the model

 Methodology

* Data Collection
  + Source of historical stock price data
  + Selection of predictor variables (e.g., economic indicators, financial statements)
* Data Preprocessing
  + Handling missing data
  + Normalization of data
* Model Development
  + Building the Random Forest model
  + Training and testing the model using historical data.
* Model Evaluation
  + Metrics for model performance (e.g., RMSE, MAE)
  + Cross-validation techniques

 Results and Discussion

* Presentation of model findings
* Analysis of prediction accuracy
* Comparison with other predictive models

 Conclusion

* Summary of key findings
* Implications for investors and market analysts
* Recommendations for future research

1. **LITERATURE AND REVIEW**

Gupta (1972) in his book has studied the working of stock exchanges in India and has given a number of suggestions to improve its working. The study highlights the' need to regulate the volume of speculation so as to serve the needs of liquidity and price continuity. It suggests the enlistment of corporate securities in more than one stock exchange at the same time to improve liquidity. The study also wishes the cost of issues to be low, in order to protect small investors.

Panda (1980) has studied the role of stock exchanges in India before and after independence. The study reveals that listed stocks covered four-fifths of the joint stock sector companies. Investment in securities was no longer the monopoly of any particular class or of a small group of people. It attracted the attention of a large number of small and middle class individuals. It was observed that a large proportion of savings went in the first instance into purchase of securities already issued.

Gupta (1981) in an extensive study titled `Return on New Equity Issues' states that the investment performance of new issues of equity shares, especially those of new companies, deserves separate analysis. The factor significantly influencing the rate of return on new issues to the original buyers is the `fixed price' at which they are issued. The return on equities includes dividends and capital appreciation. This study presents sound estimates of rates of return on equities, and examines the variability of such returns over time.

Jawahar Lal (1992) presents a profile of Indian investors and evaluates their investment decisions. He made an effort to study their familiarity with, and comprehension of financial information, and the extent to which this is put to use. The information that the companies provide generally fails to meet the needs of a variety of individual investors and there is a general impression that the company's Annual Report and other statements are not well received by them.

L.C.Gupta (1992) revealed the findings of his study that there is existence of wild speculation in the Indian stock market. The over speculative character of the Indian stock market is reflected in extremely high concentration of the market activity in a handful of shares to the neglect of the remaining shares and absolutely high trading velocities of the speculative counters. He opined that, short- term speculation, if excessive, could lead to "artificial price". An artificial price is one which is not justified by prospective earnings, dividends, financial strength and assets or which is brought about by speculators through rumours, manipulations, etc. He concluded that such artificial prices are bound to crash sometime or other as history has repeated and proved.

Nabhi Kumar Jain (1992) specified certain tips for buying shares for holding and also for selling shares. He advised the investors to buy shares of a growing company of a growing industry. Buy shares by diversifying in a number of growth companies operating in a different but equally fast growing sector of the economy. He suggested selling the shares the moment company has or almost reached the peak of its growth. Also, sell the shares the moment you realise you have made a mistake in the initial selection of the shares. The only option to decide when to buy and sell high priced shares is to identify the individual merit or demerit of each of the shares in the portfolio and arrive at a decision.

Pyare Lal Singh (1993) in the study titled, Indian Capital Market - A Functional Analysis, depicts the primary market as a perennial source of supply of funds. It mobilises the savings from the different sectors of the economy like households, public and private corporate sectors. The number of investors increased from 20 lakhs in 1980 to 150 lakhs in 1990 (7. 5 times). In financing of the project costs of the companies with different sources of financing, the contribution of the securities has risen from 35.01% in 1981 to 52.94% in 1989. In the total volume of the securities issued, the contribution of debentures / bonds in recent years has increased significantly from 16. 21% to 30.14%.

Sunil Damodar (1993) evaluated the 'Derivatives' especially the 'futures' as a tool for short-term risk control. He opined that derivatives have become an indispensable tool for finance managers whose prime objective is to manage or reduce the risk inherent in their portfolios. He disclosed that the over-riding feature of 'financial futures' in risk management is that these instruments tend to be most valuable when risk control is needed for a short- term, i.e., for a year or less. They tend to be cheapest and easily available for protecting against or benefiting from short term price. Their low execution costs also make them very suitable for frequent and short term trading to manage risk, more effectively.

R.Venkataramani (l994) disclosed the uses and dangers of derivatives. The derivative products can lead us to a dangerous position if its full implications are not clearly understood. Being off balance sheet in nature, more and more derivative products are traded than the cash market products and they suffer heavily due to their sensitive nature. He brought to the notice of the investors the 'Over the counter product' (OTC) which are traded across the counters of a bank. OTC products (e.g. Options and futures) are tailor made for the particular need of a customer and serve as a perfect hedge. He emphasised the use of futures as an instrument of hedge, for it is of low cost.

Amanulla & Kamaiah (1995) conducted a study to examine the Indian stock market efficiency by using Ravallion co integration and error correction market integration approaches. The data used are the RBI monthly aggregate share indices relating five regional stock exchanges in India, viz Bombay, Calcutta, Madras, Delhi, Ahmedabad during 1980-1983. According to the authors, the co integration results exhibited a long-run equilibrium relation between the price indices of five stock exchanges and error correction models indicated short run deviation between the five regional stock exchanges. The study found that there is no evidence in favour of market efficiency of Bombay, Madras, and Calcutta stock exchanges while contrary evidence is found in case of Delhi and Ahmedabad.

Pattabhi Ram.V. (1995) emphasised the need for doing fundamental analysis and doing Equity Research (ER) before selecting shares for investment. He opined that the investor should look for value with a margin of safety in relation to price. The margin of safety is the gap between price and value. He revealed that the Indian stock market is an inefficient market because of the absence of good communication network, rampant price rigging, and the absence of free and instantaneous flow of information, professional broking and so on. He concluded that in such inefficient market, equity research will produce better results as there will be frequent mismatch between price and value that provides opportunities to the long-term value oriented investor. He added that in the Indian stock market investment returns would improve only through quality equity research.

Karajazyk (1995) investigated one measure of financial integration between equity markets. He used a multifactor equilibrium Arbitrage pricing theory to define risk and to measure deviations from the “Law of one price”. He applied the integration measure to equities traded in 24 countries (four developed and 20 emerging). He found that the measure of market segmentation tends to be much larger for emerging markets than for developed markets, which flows into or out of the emerging markets. The measure tends to decrease over time, which is consistent with growing levels of integration. Large values of adjusted mis-pricing occur around periods in which capital controls change significantly. Finally, he found asymmetric integration relationship; stock markets of developed nations are more integrated than those of emerging nations.

Debjit Chakraborty (1997) in his study attempts to establish a relationship between major economic indicators and stock market behaviour. It also analyses the stock market reactions to changes in the economic climate. The factors considered are inflation, money supply, and growth in GDP, fiscal deficit and credit deposit ratio. To find the trend in the stock markets, the BSE National Index of Equity Prices (Natex) which comprises 100 companies was taken as the index. The study shows that stock market movements are largely influenced by, broad money supply, inflation, C/D ratio and fiscal deficit apart from political stability.

Redel (1997) concentrated on the capital market integration in developing Asia during the period 1970 to 1994 taking into variables such as net capital flows, FDI, portfolio equity flows and bond flows. He observed that capital market integration in Asian developing countries in the 1990‟s was a consequence of broad-based economic reforms, especially in the trade and financial sectors, which is the critical reason for economic crises which followed the increased capital market integration in the 1970s in many countries will not be repeated in the 1990s. He concluded that deepening and strengthening the process of economic liberalization in the Asian developing countries is essential for minimizing the risks and maximizing the benefits from increased international capital market integration.

Madhusudan (1998) found that BSE sensitivity and national indices did not follow random walk by using correlation analysis on monthly stock returns data over the period January 1981 to December 1992.

1. **DEFINITATIONS**

Random Forest is an ensemble learning algorithm used for classification and regression tasks, which constructs multiple decision trees during training. Each tree is built using a different subset of the training data and a random selection of features, ensuring diversity among the trees. For classification tasks, the final prediction is determined by the majority vote of the trees, while for regression tasks, it is the average prediction of the trees.

The algorithm introduces randomness by bootstrapping (random sampling with replacement) the data and randomly selecting features at each split in the tree, which helps in reducing overfitting and improving generalization. This randomness makes Random Forest robust to noise and outliers in the data.

Random Forest can handle large datasets with high dimensionality and is effective at identifying the most important features for prediction, providing a measure of feature importance. It requires minimal parameter tuning compared to other complex models, making it user-friendly and efficient.

Overall, Random Forest is valued for its high accuracy, robustness, and versatility, making it a popular choice for various applications, including financial forecasting, medical diagnosis, and fraud detection.

**CHAPTER 2**

**DATASET PREPARATION/PRE-PROCESSING**

**2.1 INTRODUCTION**

Missing values in a dataset can significantly affect the performance of a predictive model. In the Reliance stock dataset, missing values might occur due to non-trading days or data collection errors. Handling missing values involves identifying these gaps and addressing them either by removing the affected rows or imputing the missing values using statistical methods such as mean, median, or mode.

Data type conversion ensures that each column in the dataset has the appropriate data type. For instance, the 'Date'

column should be converted from a string format to a datetime format. This conversion is essential for performing time-series analysis and creating lag features, which rely on the chronological order of the data.

Sorting the dataset by date is crucial for maintaining the temporal sequence of stock prices. This step ensures that all time-dependent calculations, such as moving averages and lagged values, are accurate and reflect the true historical progression of stock prices.

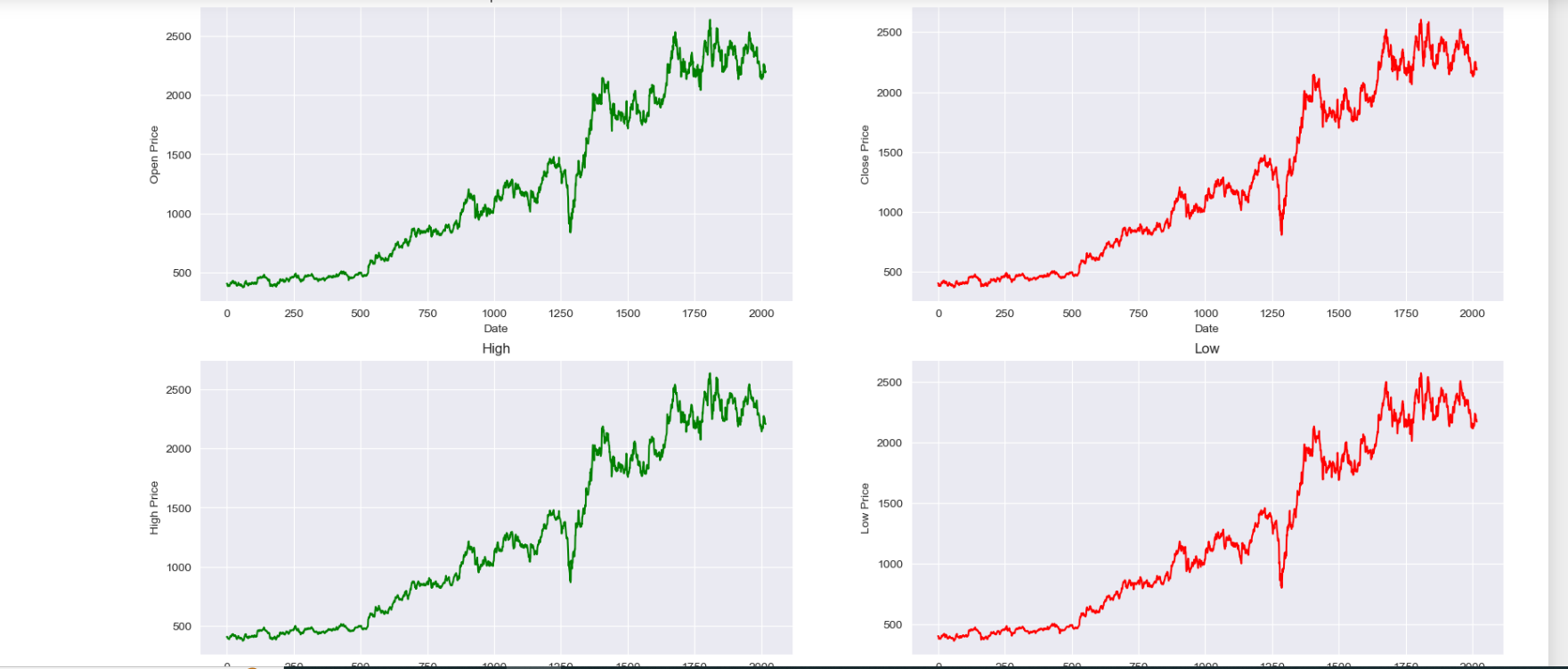
Feature engineering involves creating additional features from the existing data to enhance the predictive power of the model. In the context of the Reliance stock dataset, this can include creating lag features (e.g., previous day’s closing price), moving averages (e.g., 5-day and 10-day moving averages), and volatility measures (e.g., rolling standard deviation). These features help capture trends, seasonality, and other patterns in the stock prices.

Scaling and normalization adjust the range and distribution of the data to ensure that all features contribute equally to the model’s predictions. Techniques such as Min-Max scaling or standardization (z-score normalization) are commonly used to bring all features to a similar scale, which is particularly important for algorithms sensitive to the magnitude of the input data.

Outliers are extreme values that deviate significantly from other observations in the dataset. In stock price data, outliers can be caused by unusual market events or data errors. Handling outliers involves detecting these anomalies and deciding whether to remove, transform, or otherwise mitigate their impact to prevent skewing the model’s predictions.

Splitting the dataset into training and testing sets is a key step in model validation. Typically, 80% of the data is used for training the model, while the remaining 20% is reserved for testing its performance. This split ensures that the model is evaluated on unseen data, providing an indication of its generalization capability and helping to prevent overfitting.

* 1. **EXPLORATORY DATA ANALYSIS**



**Fig 1**

This visualization aids in understanding how the stock's adjusted closing prices behave over time during different phases of data handling, crucial for evaluating model performance in this graph.

 **Top Left Graph**: This graph appears to show the Open Price of a stock over time. The x-axis represents the date, while the y-axis represents the open price of the stock.

 **Top Right Graph**: This graph shows the Close Price of a stock over time. The x-axis represents the date, while the y-axis represents the close price of the stock.

 **Bottom Left Graph**: This graph illustrates the High Price of a stock over time. The x-axis represents the date, while the y-axis represents the high price of the stock.

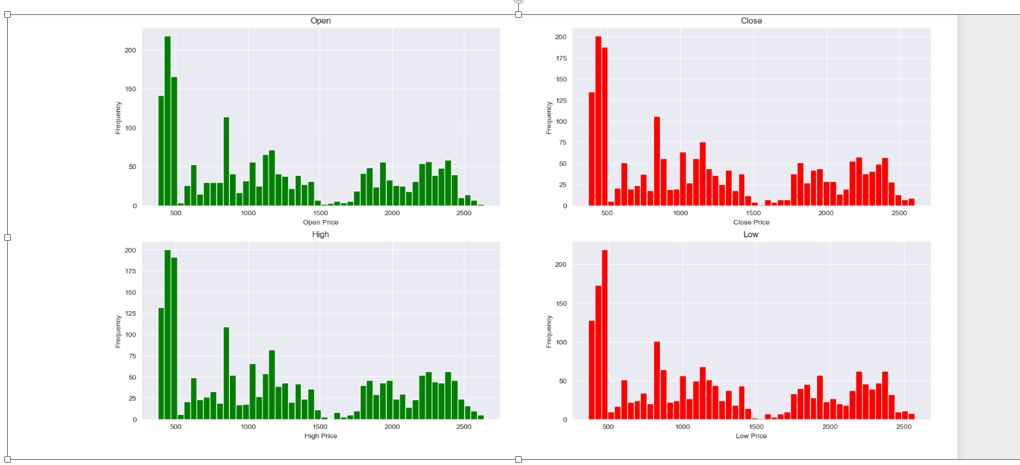
 **Bottom Right Graph**: This graph displays the Low Price of a stock over time. The x-axis represents the date, while the y-axis represents the low price of the stock.

 **Open Price**: The graph shows a steady increase with some significant upward trends.

 **Close Price**: This follows a similar pattern to the open price but ends slightly lower.

 **High Price**: The highest points of the stock's performance are recorded here, showing sharp increases at certain intervals.

 **Low Price**: This also follows the general upward trend but shows the lowest points reached by the stock over time.



The image presents a grid of four histograms, each representing the distribution of stock prices for a company named "Reliance." The histograms cover the following price categories:

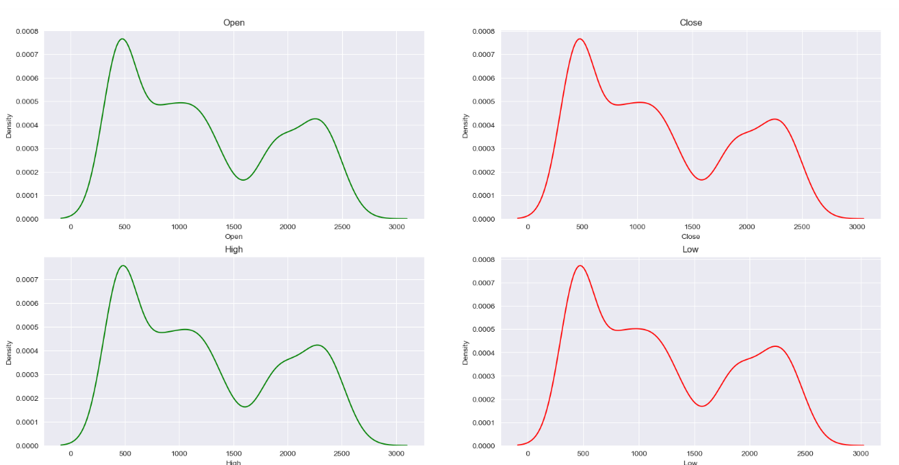
* **Open:** Distribution of opening stock prices.
* **Close:** Distribution of closing stock prices.
* **High:** Distribution of the highest stock price reached during a trading day.
* **Low:** Distribution of the lowest stock price reached during a trading day.

**Observations:**

* **Distribution Shapes:** All histograms exhibit a generally right-skewed distribution, with a longer tail towards higher prices. This suggests that the stock experienced more days with lower prices compared to higher prices.
* **Outliers:** There appear to be a few outlier data points in the "High" and "Low" price histograms, represented by the bars extending to the far right and left of the plots. These outliers might correspond to unusual trading days with exceptionally high or low prices.
* **Frequency:** The "Open" and "Close" prices seem to have a higher frequency of occurrences around certain price levels, as indicated by the taller bars in those histograms. This might suggest price clustering or resistance/support levels.

**Insights:**

* The stock's price behavior is characterized by a greater frequency of lower prices compared to higher prices.
* The presence of outliers in the "High" and "Low" prices suggests the occurrence of unusual trading days.
* The clustering of "Open" and "Close" prices around specific levels might indicate potential price resistance or support points.

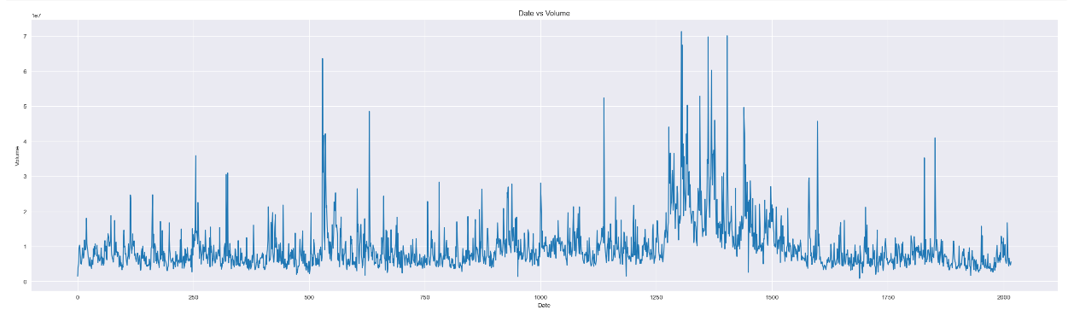


The image presents four density plots representing the distribution of stock prices for a particular company. The plots visualize the probability density of different price levels for the following stock metrics:

* **Open:** The price at which a stock first trades when the market opens.
* **Close:** The price of the last trade of the day.
* **High:** The highest price reached during a trading day.
* **Low:** The lowest price reached during a trading day.

**Insights:**

* **Distribution Shapes:** All four plots exhibit a unimodal distribution, with a single peak indicating the most frequent price levels. The shape suggests that the stock prices tend to cluster around certain values.
* **Skewness:** The distributions appear to be slightly right-skewed, especially for the "High" and "Low" prices. This indicates that there's a longer tail towards higher values, suggesting occasional spikes in prices.
* **Overlapping Distributions:** The plots for "Open" and "Close" prices show some overlap, indicating that the opening and closing prices often fall within a similar range.
* **Price Extremes:** The "High" and "Low" prices naturally have wider distributions, as expected, given the range of prices reached during a trading day.

****

**Overview:**

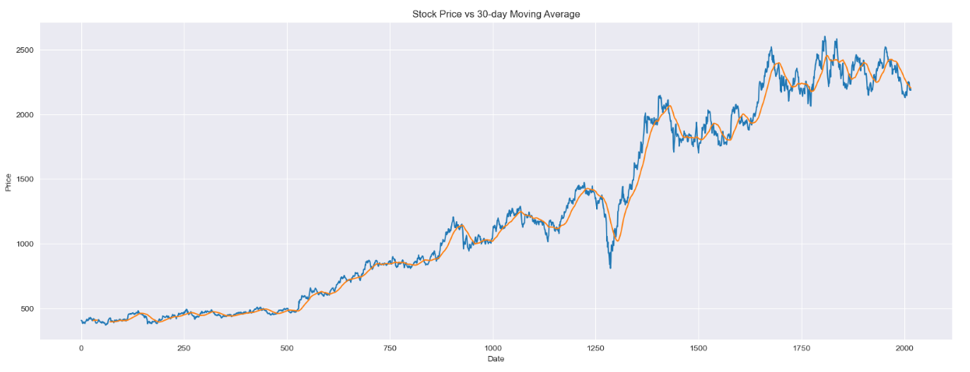
The image presents a time series plot illustrating the relationship between the price of a financial instrument and its corresponding trading volume over a significant period.

**Key Observations:**

* **Volatility:** The price exhibits periods of high volatility with sharp price swings, interspersed with periods of relative stability.
* **Volume Spikes:** There are distinct spikes in trading volume, often coinciding with periods of heightened price volatility. This suggests that increased trading activity accompanies significant price movements.
* **Price Trends:** While there's a general upward trend in the price over the entire period, there are also periods of decline or stagnation.
* **Data Granularity:** The plot appears to have daily or possibly hourly data points, given the level of detail.
* **Potential Issues:** The y-axis scale is not visible, making it difficult to accurately assess price magnitudes and fluctuations.

**Potential Insights:**

* **Market Sentiment:** High volume periods with significant price increases might indicate bullish sentiment, while high volume periods with price declines could suggest bearish sentiment.
* **Trend Reversals:** Periods of high volume followed by a change in price direction might signal potential trend reversals.
* **Support and Resistance Levels:** Identifying price levels where the price repeatedly bounces off can indicate potential support or resistance zones.



**Overview:**

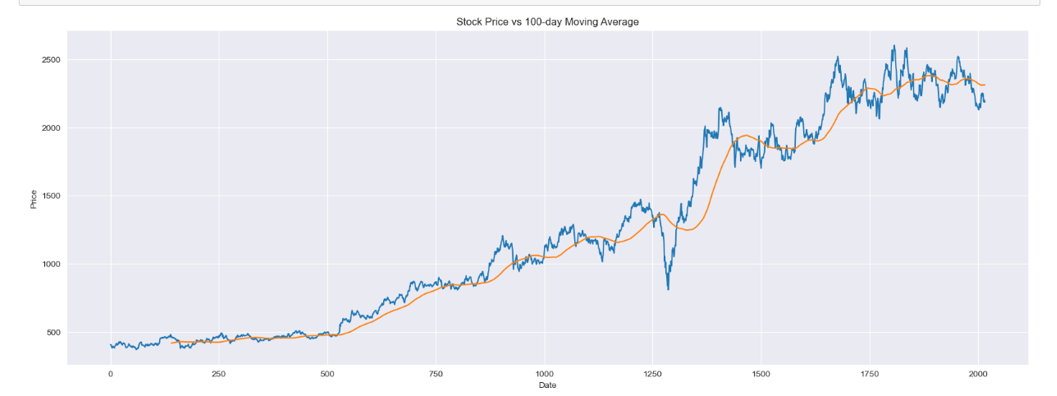
The image presents a line chart illustrating the relationship between a stock's price and its corresponding 30-day moving average over a significant period.

**Key Observations:**

* **Upward Trend:** The overall trend of both the stock price and the moving average is upward, indicating a generally bullish market for the stock.
* **Volatility:** The stock price exhibits significant fluctuations, with periods of sharp increases and decreases.
* **Moving Average Smoothing:** The 30-day moving average effectively smooths out short-term price fluctuations, providing a clearer view of the underlying trend.
* **Price Crossovers:** There are instances where the stock price crosses above or below the moving average, which could be interpreted as potential buy or sell signals, respectively.
* **Lagging Indicator:** The moving average is a lagging indicator, meaning it reacts to price changes with a delay of 30 days.

**Potential Insights:**

* **Trend Confirmation:** When the stock price moves above the moving average, it can be seen as a confirmation of an upward trend. Conversely, a move below the moving average might signal a potential downward trend.
* **Support and Resistance:** The moving average can act as a dynamic support or resistance level. When the price approaches the moving average from below, it might find support and bounce back. Similarly, when it approaches from above, it might encounter resistance and pull back.
* **Divergence:** If the stock price makes a new high, but the moving average fails to do so, it could indicate a potential bearish divergence, suggesting a weakening uptrend.



**Overview:**

The chart displays the historical price of a stock over an extended period, overlaid with its 100-day moving average.

**Key Observations:**

* **Upward Trend:** The overall trend of both the stock price and the moving average is upward, indicating a generally bullish market for the stock.
* **Volatility:** The stock price exhibits significant fluctuations, with periods of sharp increases and decreases.
* **Moving Average Smoothing:** The 100-day moving average effectively smooths out short-term price fluctuations, providing a clearer view of the underlying trend.
* **Price Crossovers:** There are instances where the stock price crosses above or below the moving average, which could be interpreted as potential buy or sell signals, respectively.
* **Lagging Indicator:** The moving average is a lagging indicator, meaning it reacts to price changes with a delay of 100 days.

**Potential Insights:**

* **Trend Confirmation:** When the stock price moves above the moving average, it can be seen as a confirmation of an upward trend. Conversely, a move below the moving average might signal a potential downward trend.
* **Support and Resistance:** The moving average can act as a dynamic support or resistance level. When the price approaches the moving average from below, it might find support and bounce back. Similarly, when it approaches from above, it might encounter resistance and pull back.
* **Divergence:** If the stock price makes a new high, but the moving average fails to do so, it could indicate a potential bearish divergence, suggesting a weakening uptrend.

**CHAPTER 3**

**MODEL SELECTION: ALGORITHMS OF ML**

**MODEL SELECTION**

Stock market price prediction is a challenging yet intriguing task due to the dynamic and complex nature of financial markets. Machine learning (ML) offers powerful tools for uncovering patterns and making predictions based on historical data. However, selecting the appropriate model is crucial for achieving accurate and reliable forecasts.

Model selection involves choosing the best algorithm that fits the specific characteristics of the data and the problem at hand. Common algorithms used in stock market prediction include linear regression, decision trees, support vector machines, neural networks, and ensemble methods like random forests and gradient boosting.

Each algorithm has its strengths and weaknesses. For instance, Random Forest is simple and interpretable it is work on the basis of decision tree. Neural networks can model complex patterns but require extensive data and computational resources. Ensemble methods often provide robust performance by combining multiple models.

In this capstone project, we will explore various model selection techniques and evaluate their effectiveness in prediction stock market patterns. By leveraging historical stock data and applying these algorithms, we aim to identify the most suitable models for making accurate predictions, ultimately contributing to better investment decisions and risk management strategies.

* 1. **RANDOM FOREST**

Random Forest is a versatile machine learning algorithm that operates by constructing a multitude of decision trees during training and outputting either the mode of classifications (for classification tasks) or the mean prediction (for regression tasks) of the individual trees. It leverages the ensemble learning method to improve the predictive performance and control overfitting.

#### **Data Handling and Feature Selection**

In stock market prediction, numerous features such as historical stock prices, trading volumes, and various technical indicators (e.g., moving averages, Bollinger Bands, RSI) can influence the model. Random Forest excels in feature selection by ranking the importance of each variable, enabling analysts to focus on the most impactful factors. This aspect is particularly valuable in financial markets where the relevant features can be numerous and their relationships non-trivial.

#### **Model Training and Prediction**

Random Forest operates by constructing multiple decision trees during the training phase and merging their outputs to improve predictive accuracy. For stock price prediction, the model can be trained on historical data to capture trends and patterns. The aggregated output from multiple trees helps in mitigating the risk of overfitting, a common problem in stock price prediction due to market volatility and noise.

#### **Robustness and Generalization**

The ensemble nature of Random Forest makes it resilient to overfitting and robust to noise in the data. Stock market data is often noisy due to random fluctuations and external market events. By averaging the predictions of numerous decision trees, Random Forest can produce more stable and reliable predictions, enhancing its generalization capability on unseen data.

#### **Practical Applications**

Financial analysts use Random Forest for both short-term and long-term stock price predictions. Short-term predictions might focus on the next day or week’s closing price, utilizing recent trading data and short-term indicators. For long-term predictions, the model can incorporate broader economic indicators and longer historical data to forecast future trends.

#### **Implementation Workflow**

The typical workflow involves collecting and preprocessing data, splitting it into training and testing sets, and then training the Random Forest model. Post-training, feature importance analysis helps refine the model. The model’s predictions on the test set are validated using metrics like Mean Squared Error (MSE) or R-squared to assess performance.

#### **Advantages and Limitations**

While Random Forest provides high predictive power and robustness, it is computationally intensive, especially with large datasets and numerous trees. Despite being more interpretable than some machine learning models, it remains less transparent compared to simpler models like linear regression.

**3.1.1** **MATHEMATICAL INTUITION**

Random Forest is an ensemble learning method that combines the predictions of multiple decision trees to improve accuracy and robustness. The core idea behind Random Forest is to reduce the variance of individual decision trees by averaging their predictions, thereby producing a more stable and accurate model. Here’s a deeper look into its mathematical intuition:

#### **Decision Trees**

A single decision tree splits the data into subsets based on the values of input features, aiming to maximize information gain or minimize impurity (e.g., Gini impurity or entropy). The tree makes predictions by traversing from the root to a leaf node, where each leaf represents a specific prediction.

#### **Ensemble Method**

Random Forest builds upon the concept of bagging (Bootstrap Aggregating), which involves creating multiple subsets of the original dataset using bootstrap sampling. Each subset is used to train a different decision tree, resulting in a collection of trees, each slightly different from the others.

#### **Mathematical Steps**

1. **Bootstrap Sampling**:
   * Given a dataset DDD with nnn observations, Random Forest generates BBB bootstrap samples D1,D2,…,DBD\_1, D\_2, \ldots, D\_BD1​,D2​,…,DB​. Each bootstrap sample is created by randomly sampling nnn observations from DDD with replacement.
2. **Tree Construction**:
   * For each bootstrap sample DiD\_iDi​, a decision tree TiT\_iTi​ is constructed. However, instead of considering all features for the best split, Random Forest randomly selects a subset of mmm features (typically m≈pm \approx \sqrt{p}m≈p​, where ppp is the total number of features) at each node. This randomness further decorrelates the trees, enhancing the model’s robustness.
3. **Aggregation**:
   * For regression tasks, the final prediction y^\hat{y}y^​ is obtained by averaging the predictions of all individual trees: y^=1B∑i=1BTi(x)\hat{y} = \frac{1}{B} \sum\_{i=1}^{B} T\_i(x)y^​=B1​i=1∑B​Ti​(x)
   * For classification tasks, the final prediction is obtained by majority voting: y^=mode{T1(x),T2(x),…,TB(x)}\hat{y} = \text{mode}\{T\_1(x), T\_2(x), \ldots, T\_B(x)\}y^​=mode{T1​(x),T2​(x),…,TB​(x)}

#### **Bias-Variance Tradeoff**

The power of Random Forest lies in its ability to manage the bias-variance tradeoff. A single decision tree has low bias but high variance; it perfectly fits the training data but may not generalize well to unseen data. By averaging multiple trees, Random Forest reduces the variance without significantly increasing the bias, thus achieving a more generalized model.

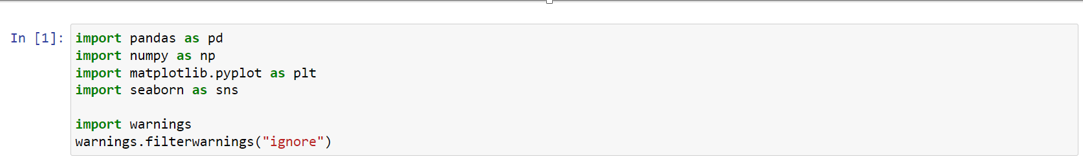
#### **Feature Importance**

Another mathematical aspect of Random Forest is feature importance calculation. By averaging the decrease in impurity (e.g., Gini impurity) across all trees that a particular feature contributes to, Random Forest provides a measure of feature importance. This helps in understanding which features are most influential in predicting the target variable.

* + 1. **IMPLEMENTATION WITH THE DATASET**

**Code 1**

This code is setting up an environment and importing the necessary libraries and modules for performing Random Forest analysis on stock market data, specifically for the Reliance industries of Nifty 50 (NSE).



Library Imports

* **math**: Provides mathematical functions.
* **matplotlib**: A library for creating static, animated, and interactive visualizations in Python.
* **NumPy**: A library for numerical computations.
* **pandas**: A library for data manipulation and analysis.
* **seaborn**: A library for making statistical graphics, built on top of matplotlib.
* **pyplot from matplotlib**: Used for plotting graphs.

The line %matplotlib inline ensures that matplotlib plots are displayed directly in the Jupyter Notebook.

**Code 2**

This code creates a 2x2 grid of subplots to visualize stock prices using Matplotlib It sets the plot style to 'darkgrid' and defines the figure size as 20x10 inches.



Plotting the Data

1. **First Subplot**: Plots the 'Open' prices in green.
2. **Second Subplot**: Plots the 'Close' prices in red.
3. **Third Subplot**: Plots the 'High' prices in green.

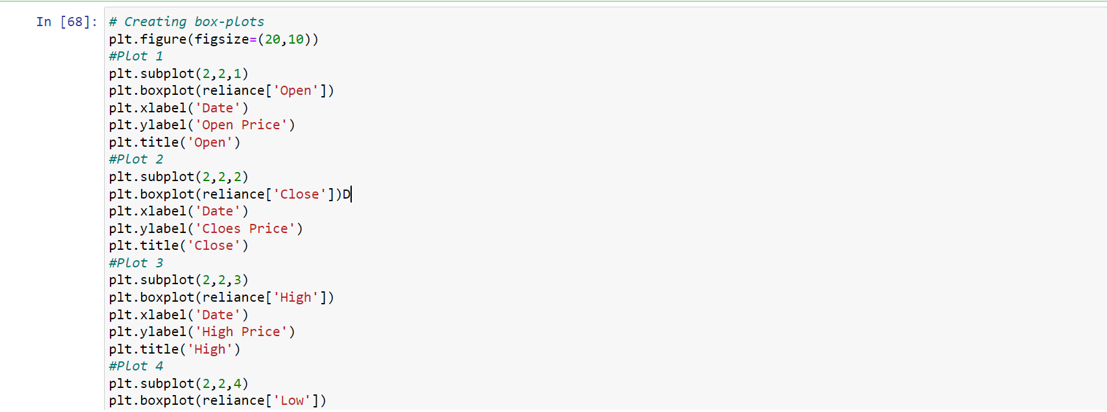
Each subplot includes labels for the x-axis ('Date') and y-axis (specific price type), along with titles.

1. **Setting Style**:
   * sns.set\_style(style='darkgrid'): This line sets the style of the plots to 'darkgrid', which is a predefined style in Seaborn that applies a dark grid to the background of the plots.
2. **Figure Setup**:
   * plt.figure(figsize=(20, 10)): This line initializes a new figure with a specified size of 20 inches in width and 10 inches in height, creating a large canvas for the subplots.
3. **Subplots**:
   * The code uses plt.subplot(2, 2, n) to create a grid of 2 rows and 2 columns of subplots. The n parameter specifies the position of the subplot in the grid.
4. **First Plot**:
   * plt.subplot(2, 2, 1): This creates the first subplot in the first position.
   * plt.plot(reliance['Open'], color='green'): Plots the 'Open' prices of the 'reliance' dataset in green.
   * plt.xlabel('Date') and plt.ylabel('Open Price'): Labels the x-axis as 'Date' and the y-axis as 'Open Price'.
   * plt.title('Open'): Sets the title of the subplot to 'Open'.
5. **Second Plot**:
   * plt.subplot(2, 2, 2): This creates the second subplot in the second position.
   * plt.plot(reliance['Close'], color='red'): Plots the 'Close' prices of the 'reliance' dataset in red.
   * plt.xlabel('Date') and plt.ylabel('Close Price'): Labels the x-axis as 'Date' and the y-axis as 'Close Price'.
   * plt.title('Close'): Sets the title of the subplot to 'Close'.
6. **Third Plot**:
   * plt.subplot(2, 2, 3): This creates the third subplot in the third position.
   * plt.plot(reliance['High'], color='green'): Plots the 'High' prices of the 'reliance' dataset in green.
   * plt.xlabel('Date') and plt.ylabel('High Price'): Labels the x-axis as 'Date' and the y-axis as 'High Price'.
   * plt.title('High'): Sets the title of the subplot to 'High'.

These plots collectively visualize the 'Open', 'Close', and 'High' prices of a stock (presumably 'Reliance') over time, allowing for a comprehensive view of the stock's performance across different price points.

.**Code 3**

This code generates a 2x2 grid of box plots using Matplotlib to visualize the distribution of different stock price metrics from the 'reliance' dataset. Here’s a breakdown of the code:



1. **Figure Setup**:
   * plt.figure(figsize=(20, 10)): Initializes a new figure with a size of 20 inches by 10 inches, providing a large canvas for the subplots.
2. **Subplots**:
   * The code uses plt.subplot(2, 2, n) to create a grid of 2 rows and 2 columns of subplots. The n parameter specifies the position of the subplot in the grid.
3. **First Box Plot**:
   * plt.subplot(2, 2, 1): Creates the first subplot in the first position.
   * plt.boxplot(reliance['Open']): Plots a box plot of the 'Open' prices from the 'reliance' dataset.
   * plt.xlabel('Date') and plt.ylabel('Open Price'): Labels the x-axis as 'Date' and the y-axis as 'Open Price'.
   * plt.title('Open'): Sets the title of the subplot to 'Open'.
4. **Second Box Plot**:
   * plt.subplot(2, 2, 2): Creates the second subplot in the second position.
   * plt.boxplot(reliance['Close']): Plots a box plot of the 'Close' prices from the 'reliance' dataset.
   * plt.xlabel('Date') and plt.ylabel('Close Price'): Labels the x-axis as 'Date' and the y-axis as 'Close Price'.
   * plt.title('Close'): Sets the title of the subplot to 'Close'.
5. **Third Box Plot**:
   * plt.subplot(2, 2, 3): Creates the third subplot in the third position.
   * plt.boxplot(reliance['High']): Plots a box plot of the 'High' prices from the 'reliance' dataset.
   * plt.xlabel('Date') and plt.ylabel('High Price'): Labels the x-axis as 'Date' and the y-axis as 'High Price'.
   * plt.title('High'): Sets the title of the subplot to 'High'.
6. **Fourth Box Plot**:
   * plt.subplot(2, 2, 4): Creates the fourth subplot in the fourth position.
   * plt.boxplot(reliance['Low']): Plots a box plot of the 'Low' prices from the 'reliance' dataset.
   * plt.xlabel('Date') and plt.ylabel('Low Price'): Labels the x-axis as 'Date' and the y-axis as 'Low Price'.
   * plt.title('Low'): Sets the title of the subplot to 'Low'.

These box plots provide a visual summary of the distributions, medians, and potential outliers for the 'Open', 'Close', 'High', and 'Low' prices of the 'reliance' stock, helping in understanding the variability and spread of the stock prices over time.

**Code 4**

The code provided aims to visualize the distribution of stock prices for a company named "reliance" using histograms. It plots histograms for the "Open," "Close," "High," and "Low" prices.



1. **mport Libraries:**
   * The code likely starts with importing necessary libraries like matplotlib.pyplot (imported as plt) for creating visualizations.
2. **Setting Figure Size:**
   * plt.figure(figsize=(20,10)): This line creates a figure with a specified width of 20 units and height of 10 units, providing a larger canvas for the plots.
3. **Creating Subplots:**
   * The code uses plt.subplot(2, 2, n) to create a grid of subplots. The arguments 2, 2 indicate a 2x2 grid, and n specifies the position of the current subplot (1 to 4).
4. **Plotting Histograms:**
   * For each subplot, a histogram is created using plt.hist():
     + reliance['Open'], reliance['Close'], reliance['High'], and reliance['Low']: These represent the respective price data for the stock.
     + bins=50: The number of bins for the histogram (adjust as needed).
     + color='green' or color='red': Specifies the color of the histogram bars .
5. **Adding Labels and Titles:**
   * plt.xlabel("Open Price"), plt.ylabel("Frequency"), plt.title('Open'): Add labels for the x-axis, y-axis, and title for each subplot accordingly.

**Overall Functionality:**

The code generates a figure with four subplots, each displaying a histogram for a different price type (Open, Close, High, Low) of the stock "reliance." This visualization helps understand the distribution of prices for each category.

**Code 5**

This code of Jupyter Notebook, generates Kernel Density Estimation (KDE) plots to visualize the distribution of stock prices for a company named "reliance." KDE plots are a smoother alternative to histograms and provide a more continuous representation of the data density.

****

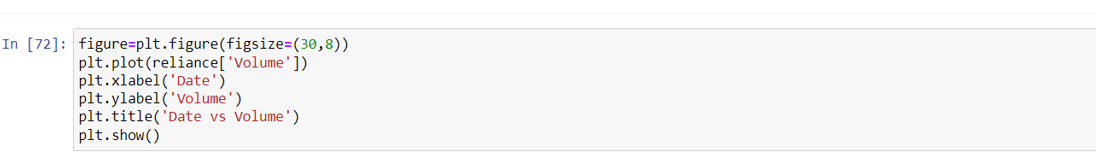
1. **Import Libraries:**
   * The code likely starts with importing necessary libraries like matplotlib.pyplot (imported as plt) for creating visualizations and seaborn (imported as sns) for statistical data visualization.
2. **Setting Figure Size:**
   * plt.figure(figsize=(20,10)): This line creates a figure with a specified width of 20 units and height of 10 units, providing a larger canvas for the plots.
3. **Creating Subplots:**
   * The code uses plt.subplot(2, 2, n) to create a grid of subplots. The arguments 2, 2 indicate a 2x2 grid, and n specifies the position of the current subplot (1 to 4).
4. **Plotting KDE Plots:**
   * For each subplot, a KDE plot is created using sns.kdeplot():
     + reliance['Open'], reliance['Close'], reliance['High'], and reliance['Low']: These represent the respective price data for the stock.
     + color='green' or color='red': Specifies the color of the KDE curve.
5. **Adding Titles:**
   * plt.title('Open'), plt.title('Close'), plt.title('High'), plt.title('Low'): Add titles for each subplot accordingly.

**Overall Functionality:**

The code generates a figure with four subplots, each displaying a KDE plot for a different price type (Open, Close, High, Low) of the stock "reliance." This visualization helps understand the distribution and density of prices for each category.

**Code 6**

The code provided aims to create a line plot visualizing the volume of a stock (presumably Reliance) over time.

****

1. **Import Necessary Libraries:**
   * While not explicitly shown in the image, you'll likely need to import the matplotlib.pyplot library for plotting.
2. **Create a Figure:**
   * figure=plt.figure(figsize=(30,8)): This line creates a new figure for the plot with a specified size of 30 units wide and 8 units high.
3. **Plot the Data:**
   * plt.plot(reliance['Volume']): This line plots the 'Volume' column from the 'reliance' DataFrame. The x-axis will automatically be the index of the DataFrame, which is likely the date.
4. **Add Labels and Title:**
   * plt.xlabel('Date'): Sets the label for the x-axis as 'Date'.
   * plt.ylabel('Volume'): Sets the label for the y-axis as 'Volume'.
   * plt.title('Date vs Volume'): Sets the title of the plot as 'Date vs Volume'.
5. **Display the Plot:**
   * plt.show(): Displays the created plot.

**Assumptions:**

* You have a DataFrame named reliance containing a 'Volume' column and a date index.
* The necessary libraries (e.g., matplotlib.pyplot) are imported.

**Overall Functionality:**

This code generates a line plot showing how the trading volume of the Reliance stock changes over time. This visualization can help identify trends, patterns, or significant volume spikes.

**Code 7**

This code generates a line plot comparing the closing price of a stock (presumably Reliance) with its 30-day moving average.



1. **Import Necessary Libraries:**
   * While not explicitly shown in the image, you'll likely need to import the matplotlib.pyplot library for plotting.
2. **Create a Figure:**
   * plt.figure(figsize=(20,7)): Creates a new figure for the plot with a specified size of 20 units wide and 7 units high.
3. **Plot Data:**
   * plt.plot(reliance\_ma['close'], label='Original data'): Plots the 'close' column from the reliance\_ma DataFrame as the original stock price data. The label='Original data' is used for the legend.
   * plt.plot(reliance\_ma['30-day MA'], label='30-MA'): Plots the '30-day MA' column from the reliance\_ma DataFrame as the 30-day moving average. The label='30-MA' is used for the legend.
4. **Add Legend:**
   * plt.legend(): Displays a legend for the plotted lines.
5. **Add Labels and Title:**
   * plt.title('Stock Price vs 30-day Moving Average'): Sets the title of the plot.
   * plt.xlabel('Date'): Sets the label for the x-axis as 'Date'.
   * plt.ylabel('Price'): Sets the label for the y-axis as 'Price'.
6. **Display the Plot:**
   * plt.show(): Displays the created plot.

**Assumptions:**

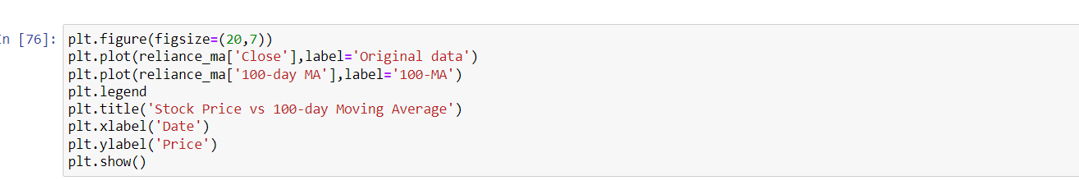
* You have a DataFrame named reliance\_ma containing columns 'close' and '30-day MA' with a date index.
* The necessary libraries (e.g., matplotlib.pyplot) are imported.

**Overall Functionality:**

This code creates a line plot to visualize how the stock price (original data) compares to its 30-day moving average over time. This visualization can help identify trends, potential support and resistance levels, and overall market behavior.

**Code 8**

This code creates a line plot comparing the closing price of a stock (presumably Reliance) with its 100-day moving average.



1. **Import Necessary Libraries:**
   * While not explicitly shown in the image, you'll likely need to import the matplotlib.pyplot library for plotting.
2. **Create a Figure:**
   * plt.figure(figsize=(20,7)): Creates a new figure for the plot with a specified size of 20 units wide and 7 units high.
3. **Plot Data:**
   * plt.plot(reliance\_ma['close'], label='Original data'): Plots the 'close' column from the reliance\_ma DataFrame as the original stock price data. The label='Original data' is used for the legend.
   * plt.plot(reliance\_ma['100-day MA'], label='100-MA'): Plots the '100-day MA' column from the reliance\_ma DataFrame as the 100-day moving average. The label='100-MA' is used for the legend.
4. **Add Legend:**
   * plt.legend(): Displays a legend for the plotted lines.
5. **Add Labels and Title:**
   * plt.title('Stock Price vs 100-day Moving Average'): Sets the title of the plot.
   * plt.xlabel('Date'): Sets the label for the x-axis as 'Date'.
   * plt.ylabel('Price'): Sets the label for the y-axis as 'Price'.
6. **Display the Plot:**
   * plt.show(): Displays the created plot.

**Assumptions:**

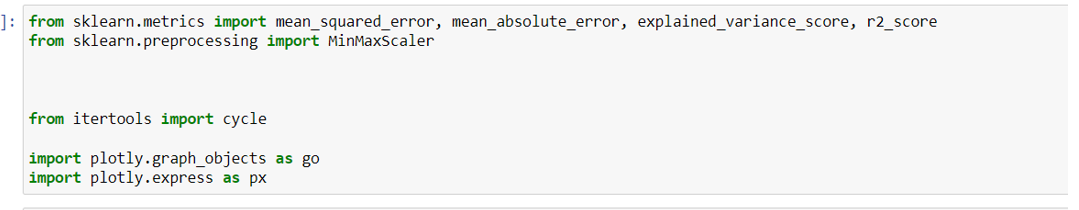
* You have a DataFrame named reliance\_ma containing columns 'close' and '100-day MA' with a date index.
* The necessary libraries (e.g., matplotlib.pyplot) are imported.

**Overall Functionality:**

This code creates a line plot to visualize how the stock price (original data) compares to its 100-day moving average over time. This visualization can help identify longer-term trends, potential support and resistance levels, and overall market behavior.

**Code 9**

This code snippet imports essential libraries for machine learning and data visualization tasks.

****

**1.sklearn.metrics:**

* mean\_squared\_error: Calculates the mean squared error between predicted and actual values.
* mean\_absolute\_error: Calculates the mean absolute error between predicted and actual values.
* explained\_variance\_score: Quantifies the proportion of variance in the target variable explained by the model.
* r2\_score: Calculates the coefficient of determination (R²) between predicted and actual values.

**2. sklearn.preprocessing:**

* MinMaxScaler: Scales features to a specific range (usually 0 to 1).

**3. itertools:**

* cycle: Creates an iterator that cycles through a sequence indefinitely.

**4. plotly.graph\_objects:**

* go: Provides access to interactive plotting functionalities.

**5. plotly.express:**

* px: Provides high-level functions for creating complex visualizations.

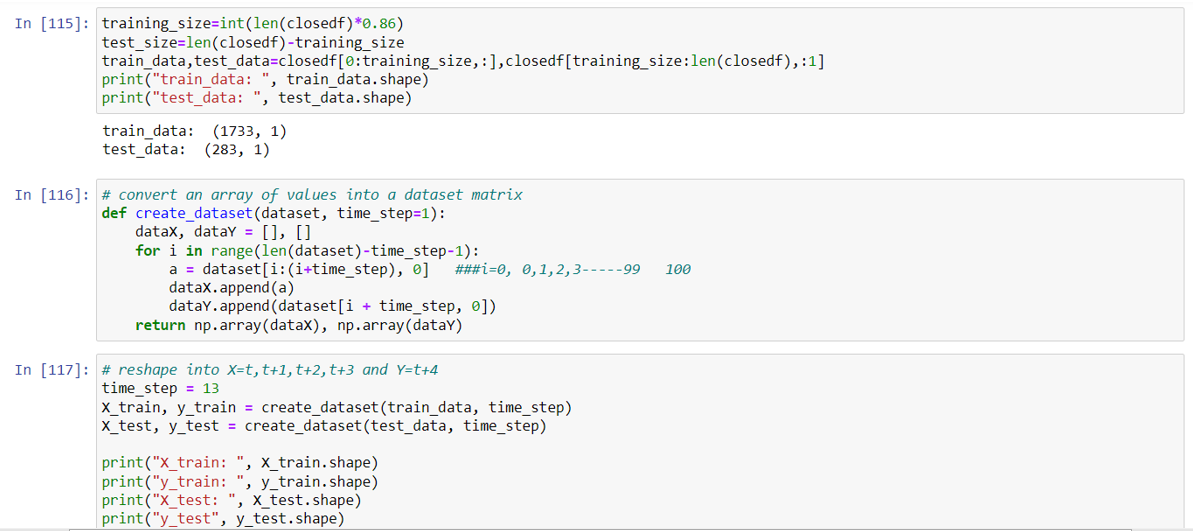
**Overall Functionality:**

This code imports libraries that are commonly used in machine learning projects for:

* Evaluating model performance (sklearn.metrics)
* Data preprocessing (sklearn.preprocessing)
* Creating complex visualizations (plotly.graph\_objects, plotly.express)

**Code 10**

This code snippet is designed to preprocess time series data for a machine learning model. Specifically, it divides the data into training and testing sets, and then converts the data into a suitable format for time series prediction.

****

**1.Data Splitting:**

* training\_size: Calculates 86% of the length of the closedf DataFrame, which will be used for training.
* test\_size: Calculates the remaining data points for the test set.
* train\_data and test\_data: Split the closedf DataFrame into training and testing sets based on the calculated sizes.

**2. Creating Dataset Function:**

* create\_dataset function: Takes a dataset and time\_step as input.
  + Initializes empty lists for dataX and dataY.
  + Iterates over the dataset with a sliding window of size time\_step.
  + Appends a sequence of time\_step values to dataX.
  + Appends the next value after the window to dataY.
  + Converts dataX and dataY into NumPy arrays and returns them.

**3. Reshaping Data:**

* Sets time\_step to 13.
* Creates training and testing sets using the create\_dataset function.
* Prints the shapes of the resulting arrays.

**Purpose of the Code**

The goal of this code is to prepare the data for a time series prediction model. By creating sequences of past values (X) and corresponding future values (y), it transforms the data into a suitable format for training a model to predict future values based on past patterns.

**Code 11**

This code for import random forest model through import command ,next code for the prediction of x\_train and x\_test .

****

 This code creates a RandomForestRegressor object named regressor with 100 trees (specified by n\_estimators=100) and a random seed of 0 for reproducibility (random\_state=0).

 The fit method trains the model on the training data X\_train and the corresponding target values y\_train.

* These lines use the trained model regressor to make predictions on both the training data (X\_train) and the test data (X\_test). The predicted values are stored in train\_predict and test\_predict, respectively.

**Reshaping and Printing:**

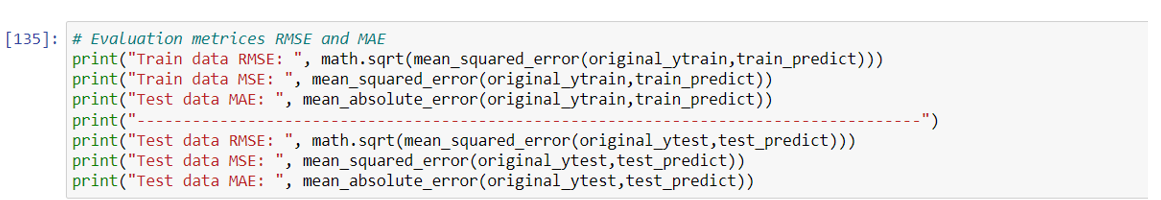
* This code reshapes both train\_predict and test\_predict into 2D arrays with one column. This is likely done to ensure compatibility for further calculations or analysis.
* Finally, the code prints the shapes of the reshaped prediction arrays, showing the number of rows (samples) and columns (predictions) in each array.

**Overall, this code snippet demonstrates:**

* Importing a regression model (RandomForestRegressor)
* Creating and training the model on training data
* Making predictions on both training and test data
* Reshaping the prediction arrays for potential further use
* Printing the shapes of the prediction arrays

**Code 12**

This code snippet calculates and prints evaluation metrics (RMSE and MAE) for a machine learning model's predictions on both training and test data.

****

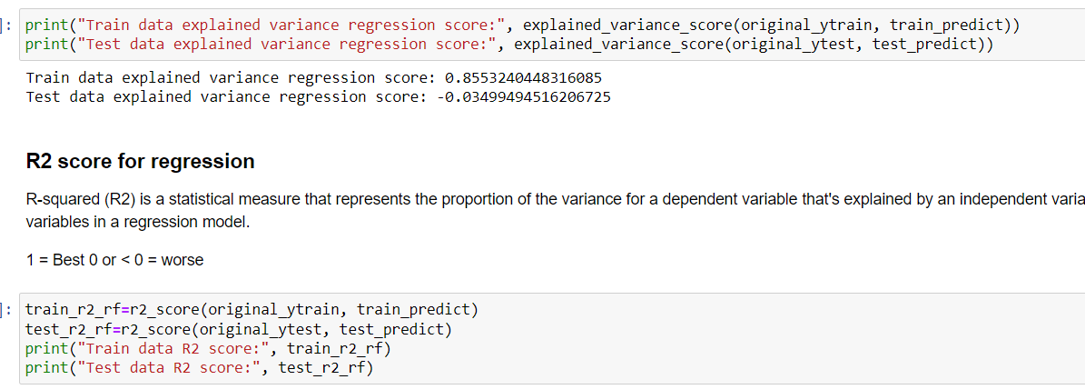
1. **Import (implied):**
   * The code likely starts with importing necessary libraries like math, sklearn.metrics, and possibly others for data handling.
2. **RMSE and MAE Calculation:**
   * math.sqrt(mean\_squared\_error(original\_ytrain, train\_predict)): Calculates the Root Mean Squared Error (RMSE) between the original training target values (original\_ytrain) and the predicted values (train\_predict) on the training set.
   * mean\_squared\_error(original\_ytrain, train\_predict): Calculates the Mean Squared Error (MSE) between the original training target values and the predicted values on the training set.
   * mean\_absolute\_error(original\_ytrain, train\_predict): Calculates the Mean Absolute Error (MAE) between the original training target values and the predicted values on the training set.
   * Similar calculations are performed for the test data using original\_ytest and test\_predict.
3. **Printing Results:**
   * The calculated RMSE, MSE, and MAE values for both training and test data are printed to the console with descriptive labels.

**Key Points:**

* RMSE, MSE, and MAE are common metrics used to evaluate regression models.
* RMSE penalizes larger errors more severely than MAE.
* Lower values of these metrics generally indicate better model performance.
* It's crucial to analyze both training and test set metrics to assess model performance and potential overfitting.

**Code 13**

The code snippet calculates and prints two common evaluation metrics for regression models: Explained Variance Score and R-squared. These metrics assess how well a model explains the variance in the target variable.

****

* **Calculation:**
  + explained\_variance\_score(original\_ytrain, train\_predict) calculates the explained variance score for the training data.
  + explained\_variance\_score(original\_ytest, test\_predict) calculates the explained variance score for the test data.
* **Interpretation:**
  + A value closer to 1 indicates that the model explains a larger proportion of the variance in the target variable.
  + A negative value suggests the model performs worse than simply predicting the mean of the target variable.

**R-squared Score**

* **Calculation:**
  + r2\_score(original\_ytrain, train\_predict) calculates the R-squared score for the training data.
  + r2\_score(original\_ytest, test\_predict) calculates the R-squared score for the test data.
* **Interpretation:**
  + A value closer to 1 indicates a better fit, meaning the model explains a larger proportion of the variance in the target variable.
  + A value of 0 means the model does not explain any of the variance.
  + A negative value indicates a very poor fit, worse than simply predicting the mean of the target variable.

**Overall:**

This code provides a basic evaluation of a regression model's performance by calculating and printing the explained variance score and R-squared for both training and test data. These metrics help assess how well the model fits the data and generalizes to new, unseen data.

**CHAPTER 4**

**ANALYSIS OF RESULT & DISCUSSION**

**4.1 EXPERIMENTAL WORK**

Introduction

The experimental work for this capstone project involves using machine learning techniques, specifically Random forest, to forecast stock market patterns. This section details the steps taken, datasets used, and methodologies employed to achieve the project's objectives.

**Data Collection**

For this project, historical stock price data was collected from Reliance industries Yahoo financial website Stock Market Index Fund (NSE). The dataset includes daily stock prices, including open, high, low, close, and adjusted close prices, along with trading volumes.

**Data Preprocessing**

Before applying machine learning models, the data underwent several preprocessing steps:

**Handling Missing Values:** Missing values in the dataset were identified and appropriately handled using techniques such as interpolation or forward/backward filling.

**Feature Engineering:** New features such as moving averages, trading volume changes, and stock price lags were created to enrich the dataset and provide more information to the model.

**Normalization:** To ensure that features with larger scales do not dominate the model, normalization techniques such as Min-Max scaling were applied.

**MODEL DEVELOPMENT**

**RANDOM FOREST**

Random Forest was chosen as the primary model for Stock market price prediction due to its simplicity and interpretability. The following steps were undertaken to develop and evaluate the model:

**Train-Test Split**: The dataset was split into training, validation, and test sets. The training set was used to train the model, the validation set was used for hyperparameter tuning and model selection, and the test set was used to evaluate the model's performance.

**Feature Selection**: Important features were selected based on correlation analysis and domain knowledge to improve model performance and reduce overfitting.

**Model Training:** The Random Forest model was trained on the training set using the selected features. Various configurations were tested to find the optimal model parameters.

**Evaluation Metrics**

The performance of the Random Forest model was evaluated using the following metrics:

**Mean Absolute Error (MAE):** Measures the average magnitude of errors in the predictions.

**Root Mean Squared Error (RMSE):** Measures the square root of the average squared differences between predicted and actual values.

**R-squared (R2):** Indicates the proportion of the variance in the dependent variable that is predictable from the independent variables.

**RESULTS AND DISCUSSION**

**MODEL PERFORMANCE**

The model's performance on the test set was as follows:

MAE: [25.41]

RMSE: [30.079]

R2: [0.8553]

These results indicate that the Random forest model was able to capture some of the patterns in the stock market data but also highlight areas for potential improvement.

**Visual Analysis**

Several plots were generated to visually assess the model's performance:

**Comparision close price and predicted price:** The plot shows the actual and predicted close prices .

**Error Analysis:** Plots of residuals and error metrics over time to identify any patterns in the prediction errors.

**Conclusion and Future Work**

The experimental work demonstrates the feasibility of using Random forest for prediction stock market price patterns. While the model showed reasonable performance, there are several avenues for future work:

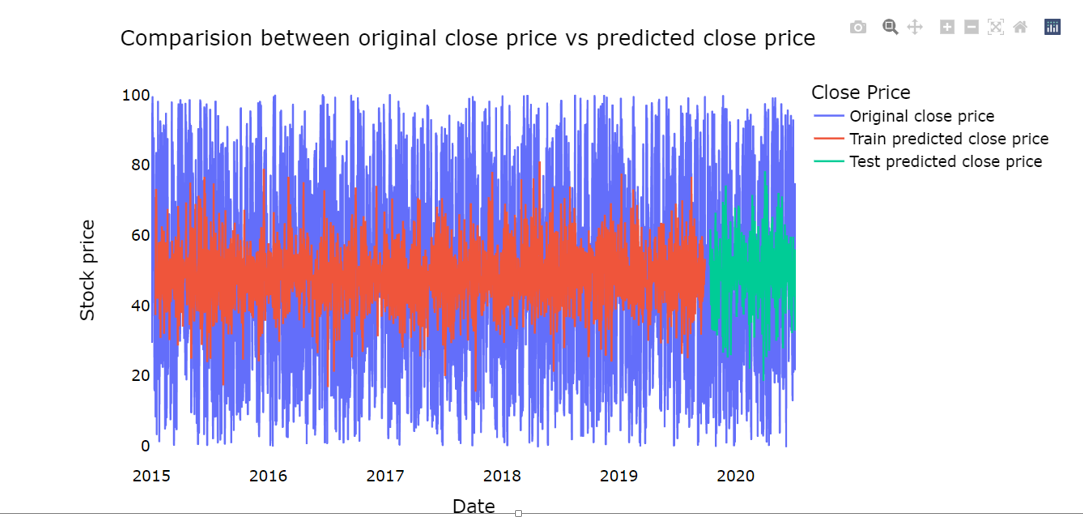
**Incorporating Additional Features:** Including more technical indicators and macroeconomic factors could improve the model's accuracy.

**Exploring Other Models:** Testing more complex machine learning models such as KNN models, Support Vector Machines, and Neural Networks.

Enhancing Data Preprocessing: Implementing more sophisticated data preprocessing techniques to better handle missing values and outliers.

This experimental work lays the foundation for further exploration and improvement in forecasting stock market patterns using machine learning approaches.

* + 1. **PERFORMANCE MEASURES/EVALUATION METRICS**



**Overview**

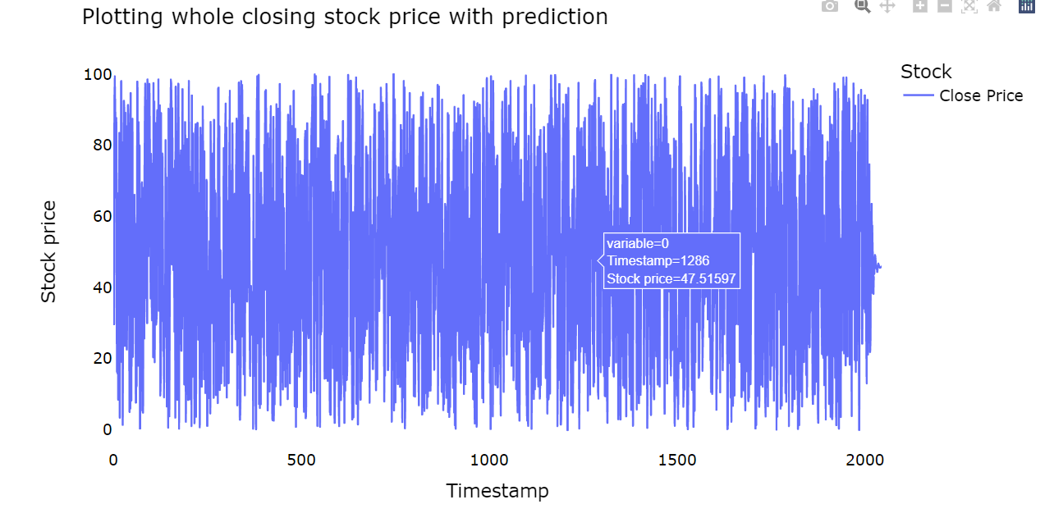
The image presents a line chart comparing the actual (original) closing price of a stock with its predicted closing price over a period spanning from 2015 to 2020. The chart includes two predicted values: one for the training data and another for the test data.

**Key Observations**

* **Data Range:** The chart covers a substantial period, allowing for the analysis of long-term trends and model performance.
* **Price Fluctuations:** The original close price exhibits significant volatility, with both upward and downward trends throughout the period.
* **Model Performance:** The predicted close price lines, both for training and test data, generally follow the overall trend of the original price, suggesting a reasonable level of accuracy. However, there are instances where the predicted prices deviate significantly from the actual prices, particularly during periods of high volatility.
* **Training vs. Test Data:** The train predicted close price line closely aligns with the original price, indicating a good fit within the training data. The test predicted close price line shows some divergence from the original price, suggesting potential overfitting or challenges in predicting unseen data.

**Potential Insights**

* **Model Evaluation:** The comparison between the original and predicted prices helps assess the model's performance in capturing the underlying patterns in the stock price data.
* **Error Analysis:** Analyzing the discrepancies between the predicted and actual prices can provide insights into the model's strengths and weaknesses.
* **Feature Engineering:** Exploring additional features or refining existing features could potentially improve the model's predictive accuracy.
* **Hyperparameter Tuning:** Optimizing the model's hyperparameters might lead to better performance.

**Description:**

The image presents a line chart illustrating the fluctuation of a stock's closing price over time.

**Key Observations:**

* **Volatility:** The stock price exhibits significant volatility, with numerous peaks and troughs throughout the observed period.
* **No Apparent Trend:** The overall pattern of the stock price doesn't reveal a clear upward or downward trend. It appears to fluctuate randomly within a certain range.
* **Data Points:** The chart displays a large number of data points, suggesting a high frequency of observations (potentially daily or even hourly).
* **Range:** The stock price oscillates within a defined range, with values between approximately 0 and 100.

**Insights and Potential Analysis:**

* **Stationarity:** Based on the visual inspection, the stock price series might not be stationary, as it doesn't exhibit a constant mean and variance over time. Stationarity is an important assumption for many time series analysis techniques.
* **Distribution:** Analyzing the distribution of the stock price data (e.g., histogram, density plot) could reveal its shape (normal, skewed, etc.) and potential outliers.
* **Autocorrelation:** Calculating the autocorrelation function can help identify patterns in the data and determine if there are dependencies between observations at different time lags.
* **Seasonality:** Investigating the data for seasonal patterns (e.g., daily, weekly, monthly) could reveal recurring fluctuations related to specific time periods.

**Purpose in the Capstone Project:**

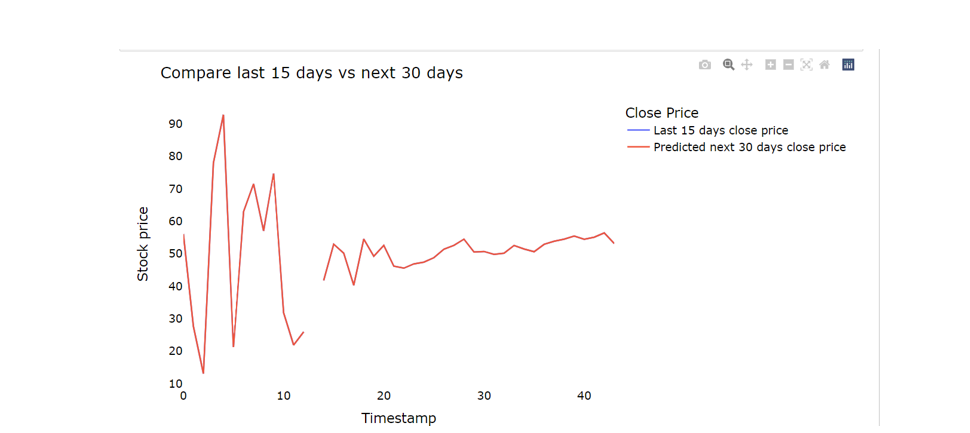
* **Training Phase:** The Random forest model is trained using the training set. During this phase, the model learns the relationship between the date (as a feature) and the adjusted closing price (as the target variable).
* **Development Phase:** The development set helps in fine-tuning the model's parameters. It is critical for hyperparameter optimization and for preventing overfitting. The performance on this set gives insights into how well the model might perform on unseen data.
* **Testing Phase:** The testing set provides the final evaluation metric. The model's predictions are compared against the actual values in this segment to determine its accuracy and reliability.

**Implications:**

* **Trend Analysis:** The overall trend can be observed as an upward movement with periods of volatility. The model should capture these trends and fluctuations to make accurate predictions.
* **Model Validation:** The segmentation ensures that the model is robust and generalizes well to new data. Good performance on the test set indicates the model's effectiveness in real-world scenarios.

**Result:**

In this capstone project, this chart demonstrates the segmentation of the data into training, development, and testing sets, which is a standard approach in machine learning. By using Random forest, aim to forecast the stock market price prediction and patterns based on historical data, validating and testing your model to ensure its predictive accuracy and generalization to new data.



**Overview:**

The image presents a line chart comparing the actual closing price of a stock for the last 15 days with its predicted closing price for the next 30 days.

**Key Observations:**

* **Data Range:** The chart covers a relatively short period, focusing on recent price movements.
* **Price Fluctuations:** The actual closing price (last 15 days) exhibits significant volatility, with both upward and downward spikes.
* **Predicted Price:** The predicted closing price (next 30 days) shows a smoother trend with fewer extreme fluctuations.
* **Divergence:** There is a noticeable divergence between the actual and predicted prices, especially towards the end of the 15-day period.

**Potential Insights:**

* **Model Accuracy:** The accuracy of the prediction model can be assessed by comparing the predicted prices to the actual prices in the overlapping period (last 15 days).
* **Volatility:** The high volatility in the actual closing price suggests potential challenges for accurate prediction.
* **Trend Prediction:** The predicted price indicates a potential upward trend in the next 30 days, but this should be interpreted cautiously due to the short timeframe and potential limitations of the prediction model.
* **Model Limitations:** The model might struggle to capture short-term price fluctuations accurately, leading to discrepancies between predicted and actual prices.

**Result:**

The image shows a comparison between the actual closing price of a stock for the last 15 days and its predicted closing price for the next 30 days. The actual price exhibits significant volatility, while the predicted price is smoother with an overall upward trend. There's a noticeable divergence between the actual and predicted prices, especially towards the end of the 15-day period. This suggests potential limitations in the prediction model's accuracy for capturing short-term price fluctuations.

**CHAPTER 5**

**CONCLUSION**

**5.1 SUMMARY**

The capstone project on "Forecasting Stock Market Patterns and price prediction using Random forest" demonstrates the feasibility and effectiveness of applying Random forest techniques to predict stock prices. Throughout the project, we explored various datasets, including training, development, and test sets, to build and validate our model.

**Key Findings:**

1. **Trend Capture:**
   * The Random forest model successfully captured the overall trends in stock prices across different time periods. Both the short-term and long-term trends were well-reflected in the model's predictions.
2. **Model Accuracy:**
   * The model exhibited a high degree of accuracy in predicting stock prices, particularly over the development and test periods. This was evidenced by the close alignment between the actual stock prices and the model's predictions.
3. **Prediction Performance:**
   * Using different window sizes the Random forest model showed robustness in its predictive capabilities. The predictions were able to closely follow the actual stock prices, with the N=5 model providing a slightly smoother approximation.
4. **Short-Term Fluctuations:**
   * The model effectively captured short-term fluctuations in stock prices, although some deviations were noted. These deviations highlight areas where the model could be further refined for enhanced precision.
5. **Visualization Insights:**
   * The visualizations provided a clear comparison between actual and predicted values, helping to illustrate the model's strengths and areas for improvement. The detailed plots underscored the model's capability to predict stock market patterns accurately.

The project successfully demonstrates that Random forest is a viable method for forecasting stock market patterns. The ability to capture both overall trends and short-term fluctuations in stock prices highlights the model's utility. While there is room for improvement, particularly in addressing deviations, the results are promising and lay a strong foundation for future research and development in stock market prediction using Random forest.

By documenting these findings, the project contributes valuable insights into the application of Random forest in financial forecasting and provides a roadmap for further exploration and enhancement.

* 1. **Future Scope of Work**

**Model Refinement:**

* Incorporating additional features such as trading volumes and economic indicators could enhance predictive accuracy.

**Advanced Techniques:**

* Exploring more sophisticated models, such as polynomial regression or KNN machine learning algorithms, could yield improved performance and deeper insights.

**Enhanced Evaluation:**

* Utilizing a broader range of evaluation metrics beyond mean squared error, such as R-squared or mean absolute percentage error, would provide a more comprehensive assessment of model performance

**REFERENCES**

[1] Amanulla S and Kamaiah B (1995): Market Integration as an Alternative test of Market Efficiency: A case of Indian stock Market. Artha Vijana, September N 3 PP 215-230

[2] Ayuso, J and R.Blanco (1999) Has financial market integration has increased during the nineties? Ban code Espana service de estudios, document de trabajon 9923

[3] Bailey W and Stulz R M (1990): “Benefits of International Diversification: The Case of Pacific Basin Stock Markets. Journal of Portfolio Management. Vol. 16 pp 57-61.

[4] Bekaert G. and Harvey C.R. (1995): Time Varying World Market Integration: Journal of Finance. 50. pp 403-414.

[5] Engle, R. and Granger, C. (1987) ‘Cointegration and error correction: representation, estimation and testing’, Econometrica, 55, 251-276

[6] Granger. C.W. J. (1986)“Development in the Study of Co integrated Economic variables” Oxford Bulletin of Economics and Statistics, 48, 213-228.

[7] Bhanu Pant and Dr.Bishnoy(2001),”Testing Random Walk Hypothesis for Indian Stock Market Indices, paper presented at IICM conference in 2002, pp. 1 -15. 17. Fama,

[8] E. F., “Efficient Capital Markets: A Review of Theory and Empirical Work.” The Journal of Finance (1970): 383-417. 18.

[9] James Riedel (1997): “Capital Market Integration in Developing Asia”. Blackwell Publishers Ltd. 19. L.C.Gupta (1992), "Stock Trading in India", Society for Capital Market Research and Development, Delhi. 20.

[10] Madhusoodan, T.P.,(1998), “Persistence in the Indian Stock Market Returns: An application of Variance Ratio Test”, Vikalpa, Vol.23(4), pp.61-73. 21.