Programme Name: MBA

Course Name: MLA-2

Course Code: MBA542B

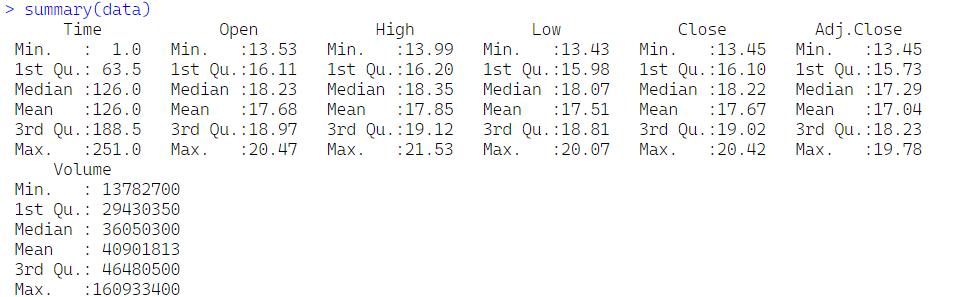
Name of the student: Harish G D

Register Number: 2327422

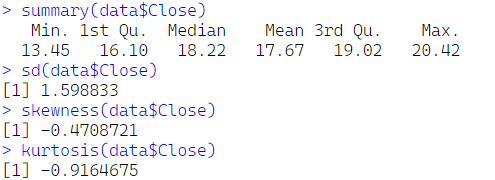
**Data Exploration:**

**Descriptive Analytics :** **:(Here you are required to discuss univariate data analysis supported to numbers and graphs with brief explanation)**

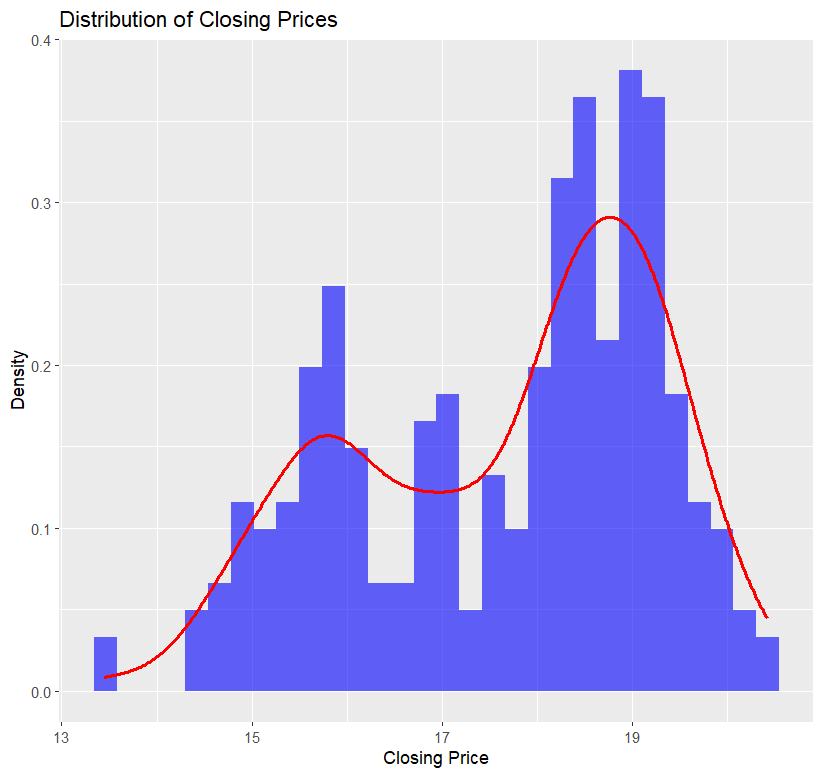
**Summary Statistics:** The Summary Statistics provide a overview about the dataset , this is useful in finding distribution of the dataset as well as the basic stats like Mean, Median and Mode.



As it was mentioned we are required to consider only the closing price in the data it is chosen as the Target variable for the prediction.



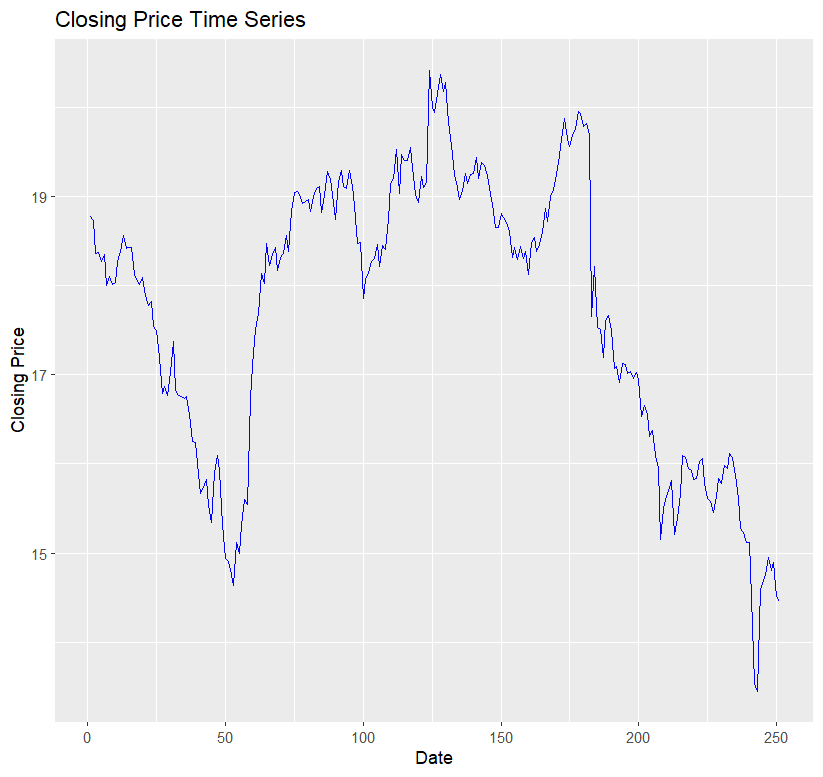
The closing prices have a relatively small range (13.45 to 20.42) and a moderate standard deviation. The distribution is slightly left-skewed, indicating some lower closing prices. The platykurtic shape suggests that there are fewer extreme values (outliers) than in a normal distribution.



**Overall Interpretation:**

The histogram provides a visual representation of the summary statistics calculated earlier. The distribution of closing prices is roughly normal, with a slight left skew and lighter tails than a perfect normal distribution.

* The mode (most frequent value) appears to be around 19.
* There are a few outliers below 14 and above 20.



**Observations:**

1. **Trend:** The overall trend of the closing prices appears to be slightly upward over the entire time period. However, there are periods of both upward and downward trends within this overall upward trend.
2. **Volatility:** The closing prices exhibit a moderate level of volatility. There are periods of high volatility with sharp price movements, as well as periods of lower volatility with more gradual price changes.
3. **Seasonality:** There doesn't appear to be any clear seasonal pattern in the data. The closing prices don't seem to follow a regular cycle that repeats itself over time.

**Feature Engineering :** **:(Here you are required to briefly discuss strategies as to how you achieved feature engineering and arrived at seven features. Also, name the response and the predictors. Print the snapshot of the final dataframe used, all the columns and 10 rows)**

**1. Log Returns :**

Log returns are commonly used in financial analysis because they are time additive and better suited for statistical modeling of price series. The formula used here calculates the difference in the logarithmic values of the closing prices, which represents the rate of return of the stock between two consecutive days.

A graph showing a number of numbers

Description automatically generated with medium confidence

**2 & 3. 10 and 50 Day Simple Moving Average**

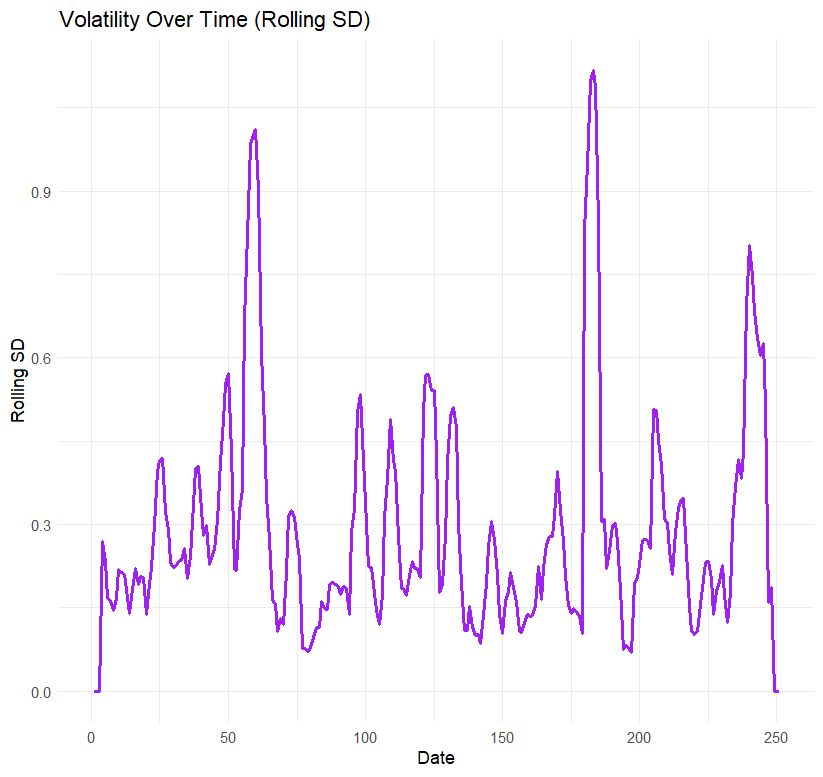
* The Simple Moving Average (SMA) is the average of the closing prices over a specified period. In this case, it’s a **10-day SMA**, which takes the average of the last 10 closing prices for each day.
* The 50-day SMA is used to gauge the longer-term market trend. It is a popular indicator to spot significant reversals or continuation of a trend, especially in conjunction with shorter-term SMAs (like the 10-day SMA).

A graph showing a price

Description automatically generated

**4. Rolling Standard Deviation (Volatility):**

This calculates the rolling standard deviation over a 7-day window. Standard deviation is a measure of volatility, showing how much the stock price fluctuates over the period. High volatility typically signals uncertainty or risk in the market. Monitoring rolling volatility is useful for identifying periods of high market uncertainty or for risk management.



**5.Momentum:**

Momentum measures the change in stock price over a certain period. This specific momentum feature calculates the difference between the current day’s closing price and the closing price 7 days ago. Positive momentum indicates an increase in price, suggesting a bullish trend, while negative momentum indicates a decrease in price, suggesting a bearish trend. It helps in predicting if a stock will continue moving in the same direction. A graph showing a number of orange lines

Description automatically generated

6. **Relative Strength Index (RSI) :**

The RSI is a momentum oscillator that measures the speed and change of price movements. It compares the magnitude of recent gains to recent losses, typically over a 14-day period.The RSI is used to identify overbought or oversold conditions in a stock. If the RSI is above 70, the stock may be overbought; if it’s below 30, it may be oversold. RSI is widely used to detect potential reversal points in the market.

A graph showing a green line

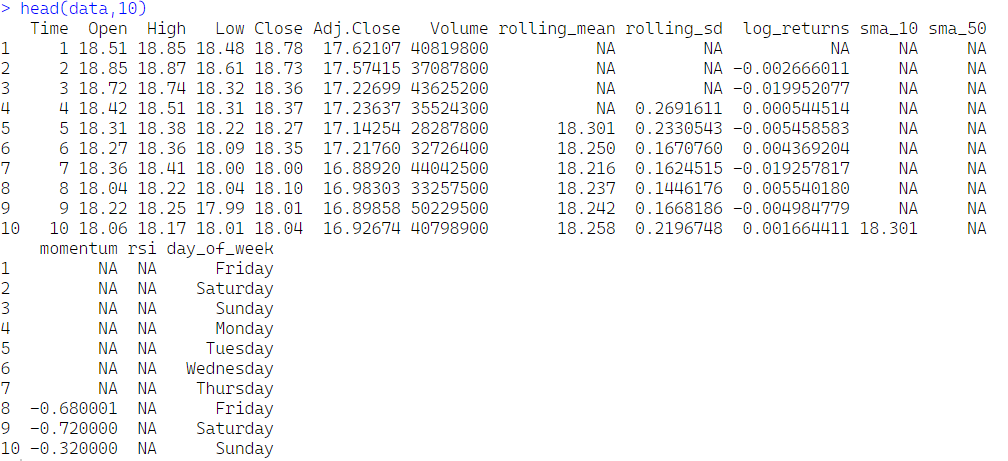
Description automatically generated

**7. Day of the Week:**

This feature extracts the day of the week from the Time column. It converts the timestamp into a factor variable representing the day of the week (e.g., "Monday", "Tuesday", etc.). Stock prices can exhibit patterns depending on the day of the week. For example, prices might fluctuate more on Mondays or Fridays, or there could be a certain market behavior depending on the day of the week. This feature can be used to capture such seasonal effects.

A graph with blue squares

Description automatically generated



**PCA Analysis :**

Principal Component Analysis (PCA) is commonly used in data analysis and machine learning for dimensionality reduction and feature extraction. The principal components highlight which variables are most influential, helping to focus on the key features.

A graph with a line

Description automatically generatedA graph with numbers and lines

Description automatically generated with medium confidence

* The variables **Open**, **Adj.Close**, and **SMA\_50** are strong drivers of variability and may be critical features.
* Correlated variables (e.g., Open and Adj.Close) might provide redundant information and could be considered for dimensionality reduction.
* Weak contributors (e.g., variables near the center) have minimal influence on the variance captured by the first two PCs.

**Model Building: (Here you will be building the XGBoost regressor and make predictions. Model summary to be printed here with loss functions preferable MSE)**

XG Boost Regressor:

**XGBoost**is a popular machine learning algorithm and it stands for **“Extreme Gradient Boosting.”** XGBoost is available in various programming languages, including **R**. An XGBoost is a fast and efficient algorithm. XG Boost works only with numeric variables. and XGBoost is a fast and efficient algorithm and is used by winners of many machine learning competitions. XG Boost works only with numeric variables. It is widely used for both classification and regression tasks.

Builiding XG Boost Regressor Model With Selected Variables:

**Key Features Selected:**

* **Open, High, Adj.Close:** Standard technical indicators related to the stock’s opening, highest, and adjusted closing prices.
* **momentum:** This feature likely measures the rate of change of the stock price, indicating trends.
* **rsi:** The Relative Strength Index is a popular momentum oscillator that shows whether a stock is overbought or oversold.
* **sma\_10 and sma\_50:** Simple Moving Averages for 10 and 50 days, widely used to smooth price data and identify trends.
* **Time:** A timestamp feature, though not used for prediction, may be relevant for tracking stock movements over time.

A screenshot of a computer code

Description automatically generated

**Model Evaluation:**

* **Predictions:** The trained model is used to predict stock prices on the test data.
* **Metrics:** Two metrics are calculated for evaluation:
  + **MAE (Mean Absolute Error) 0.129786:** This metric measures the average absolute difference between the predicted and actual values. A lower MAE also indicates better model accuracy.
  + **RMSE (Root Mean Squared Error) 0.1768:** This metric measures the average deviation of the predicted values from the actual values. A lower RMSE indicates better model accuracy..
  + **MSE (Mean Squared Error)0.03128:** This metric calculates the average squared difference between the predicted and actual values. It is similar to RMSE but squares the errors, giving more weight to larger errors.

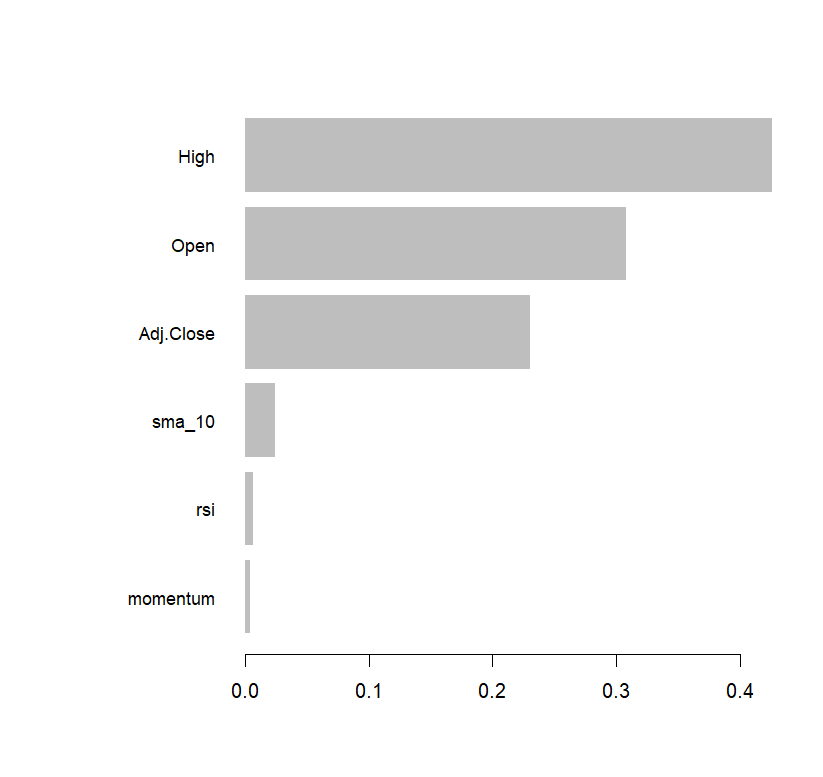
A screenshot of a computer code

Description automatically generated

Based on these metrics, the XGBoost regressor model appears to be a reasonably accurate model for predicting the closing prices. The RMSE, MAE, and MSE values are relatively small, indicating that the model's predictions are generally close to the actual values.

**Variable Importance:**

* The importance of each feature used in the model is calculated using xgb.importance() and visualized with xgb.plot.importance(). This helps to understand which features most influence the model's predictions.



**Interpretation:**

1. **High Importance:** The features "High" and "Open" have the highest importance. This suggests that these features are the most influential in determining the model's predictions. It's likely that these features are strongly correlated with the target variable (which is not explicitly shown in the plot).
2. **Moderate Importance:** The feature "Adj.Close" has moderate importance. This indicates that it also plays a significant role in the model's predictions, but to a lesser extent than "High" and "Open".
3. **Low Importance:** The features "sma\_10", "rsi", and "momentum" have very low importance. This suggests that these features have minimal impact on the model's predictions.

**Predicted vs. Actual Plot:** A line plot is created to visualize the predicted and actual stock prices over time. The actual values are shown in blue, while the predicted values are shown in red (dashed line). This helps in visually assessing the model’s performance over time.

A graph showing the difference between a graph and a chart

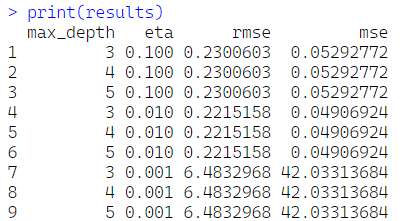
Description automatically generated

* The model tends to overestimate the actual values during periods of upward trends and underestimate them during periods of downward trends.
* The model struggles to capture the sharp price movements and tends to smooth out the fluctuations.

**Overall Interpretation:**

The plot suggests that the XGBoost model is able to capture the overall trend and major movements in the stock prices. However, it may struggle to accurately predict sharp price movements and may introduce some smoothing in the predictions.

**Experimental Designs: (Here you will be printing the nine experiments you have conducted and report the loss/cost)**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | **Depth** | **Learning Rate** | **rmse** | **mse** |
| XGBoost | 3 | 0.1 | 0.23006 | 0.052928 |
| XGBoost | 4 | 0.1 | 0.23006 | 0.052928 |
| XGBoost | 5 | 0.1 | 0.23006 | 0.052928 |
| XGBoost | 3 | 0.01 | 0.221516 | 0.049069 |
| XGBoost | 4 | 0.01 | 0.221516 | 0.049069 |
| XGBoost | 5 | 0.01 | 0.221516 | 0.049069 |
| XGBoost | 3 | 0.001 | 6.483297 | 42.03314 |
| XGBoost | 4 | 0.001 | 6.483297 | 42.03314 |
| XGBoost | 5 | 0.001 | 6.483297 | 42.03314 |

**Recommendations:**

**Key Observations**

1. **Feature Engineering and its Implication on EMH:**
   * **Log Returns, Moving Averages, and Volatility:** The calculation of features such as **log returns**, **moving averages (10-day and 50-day)**, and **volatility (rolling standard deviation)** aims to model trends and market volatility.
   * **Momentum and RSI:** Momentum indicators like **momentum** and **Relative Strength Index (RSI)** attempt to capture short-term price movements, but under EMH, they are expected to have little to no predictive value since prices follow a random walk.
2. **PCA Analysis:**
   * Dimensionality Reduction: Principal Component Analysis (PCA) is used to identify the most influential features in predicting the stock price. However, under the EMH framework, PCA should ideally not reveal any meaningful relationships, as stock prices are assumed to reflect all available information at any given time.
   * Feature Redundancy: Given the univariate nature of the dataset (only closing price), PCA’s ability to reduce dimensionality may not significantly improve model performance, further aligning with the idea that stock prices do not contain exploitable patterns.
3. **XGBoost Regressor:**
   * Model Evaluation: The XGBoost regressor is trained on the available features (timestamp and closing price) to predict future stock prices. However, under EMH, the success of this model should be questioned, as any patterns learned by the model could be mere random fluctuations rather than reliable signals.
   * Error Metrics: The RMSE (Root Mean Squared Error) and MSE (Mean Squared Error) values calculated during model evaluation reflect the predictive accuracy of the XGBoost model. While the model may provide reasonable accuracy on the historical data, under EMH, these errors should not be seen as an indication of true predictive power but rather the random nature of price movements.
4. **Model Evaluation and EMH Interpretation:**
   * Performance of XGBoost: The model’s performance metrics (RMSE = 0.221, MSE = 0.049) indicate that the XGBoost regressor can somewhat predict the closing price based on historical data. However, according to EMH, these results may not indicate any real predictability. The model may simply be fitting to noise or random patterns in the data.
   * Overfitting and Generalization: The results may suggest that the model is overfitting to past data. Since EMH argues that past prices do not offer any insight into future price movements, any model performing well on historical data should be considered suspect from an EMH standpoint, as it may not generalize to future data.

**Key Findings from the Experiments:**

1. **Optimal Hyperparameters:**
   * Depth 3 or 4 with Learning Rate of 0.01 provide the best results with the lowest RMSE and MSE, suggesting a balance between model complexity and fitting ability.
   * Learning rate of 0.1 leads to a suboptimal fit due to a potential overfitting risk, as shown by consistent RMSE and MSE values across different depths.
   * Learning rate of 0.001 results in drastic deterioration of performance, highlighting the model's underfitting tendency.
2. **EMH Considerations:**
   * The fact that the model performs reasonably well on training data may be viewed as evidence of overfitting to past noise rather than a real predictive power. Under EMH, these results do not validate the ability to predict future stock prices with any consistency.
   * Since EMH suggests stock price movements are random and reflect all known information, the prediction made by any model can only be seen as coincidental rather than genuinely predictive.
3. **Interpretation of Results:**
   * Although the model performs well, the results are unlikely to provide significant predictive value in real-world trading. Under EMH, any success in prediction is likely due to the randomness of stock price movements, making the predictions unreliable for future prices.
   * In practical terms, stock prices may follow a random walk, and future prices are as unpredictable as a coin flip, which implies that technical analysis, including momentum indicators and moving averages, may offer little advantage in predicting future prices.

**Final Recommendations:**

1. Embrace Random Walk Theory: The model evaluation suggests that stock price movements are largely unpredictable. Hence, proponents of EMH would argue that efforts to predict future prices using historical data, moving averages, or other technical indicators are futile.
2. Future Work: For more accurate stock price prediction, alternative approaches like macro-economic factors, sentiment analysis, or incorporating machine learning algorithms that can learn from broader data may offer some insight. However, under EMH, predicting stock prices with historical data alone is inherently unreliable.
3. Focus on Long-Term Trends: Rather than attempting short-term predictions, focus on long-term investment strategies based on broader economic and fundamental analysis, as EMH would suggest that short-term price movements are largely unpredictable.