Title: Detailed Analysis of Historical Stock Prices of Nvidia

Author: Girish Dommaraju

Course: CSE 587 - Data Intensive Computing

Homework 1

Abstract

This report delves into an exhaustive analysis of Nvidia's historical stock prices from July 2022 to July 2024. It outlines the processes for data cleaning, feature engineering, data visualization, and the application of machine learning techniques, focusing on clustering and predictive modeling to trends and forecast future stock movements.

Introduction

This analysis applies data-intensive computing techniques to Nvidia's stock price dataset, using Python in a Jupyter Notebook with libraries such as pandas, matplotlib, seaborn, and scikit-learn. The project aims to provide insights into stock price trends and predict future movements through machine learning

Part 1: Big Data Processing

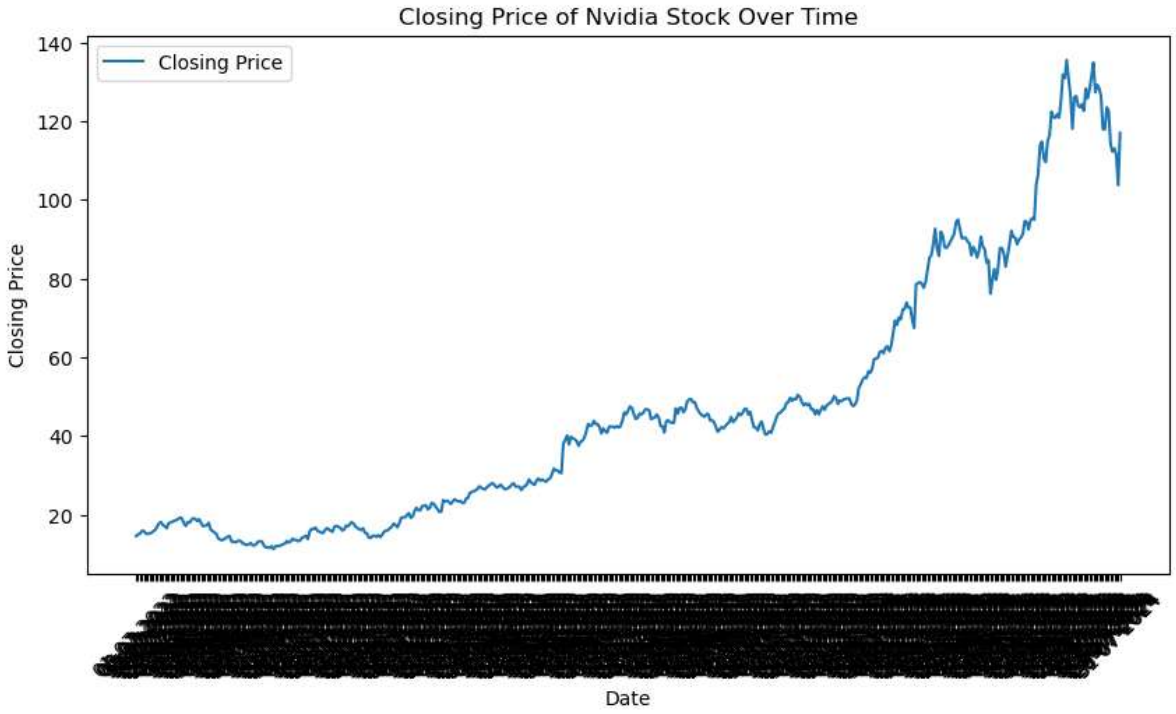
Task 1: Data Cleaning and Exploration

Data Loading and Initial Inspection: The dataset was loaded into a pandas DataFrame in the Jupyter Notebook, providing a detailed snapshot of daily trading activities, including Open, High, Low, Close, Adjusted Close, and Volume. The initial inspection identified missing values across several columns.

Cleaning Process:

- **Handling Missing Values:** Missing data points were identified and rows containing them were removed to ensure data integrity and accuracy for analysis.
- **Date Conversion:** To standardize temporal data, the 'Date' column was transformed from string to datetime format and reformatted to 'dd-mm-yyyy'.
- **Statistical Summary:** Using df.describe(), we computed and reviewed descriptive statistics for numerical columns to understand the distribution and central tendencies. This was taken for all the features.

Visualization of Cleaning Results: A line plot of closing prices over time was used to visualize trends, volatility, and the effectiveness of the cleaning process.



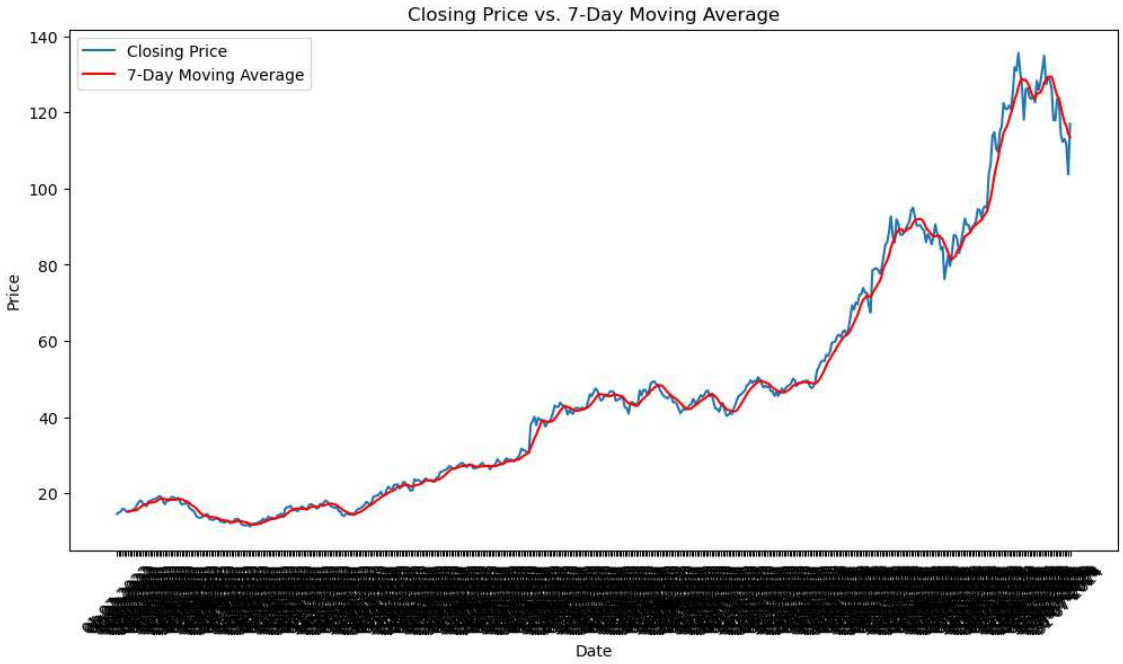
Task 2: Feature Engineering

- **Daily Returns:** This feature, calculated from the adjusted close prices, represents the percentage change in stock price from one day to the next. It is a vital indicator of the stock's short-term volatility and is often used in predictive models to forecast price movements.
- **7-Day Moving Average:** This feature smooths out short-term fluctuations and highlights longer-term trends in the stock price, providing a clearer picture of the market direction. It is particularly useful in identifying cyclical patterns and for generating trading signals.
- **Normalized Trading Volume:** Normalization of trading volume using Min-Max scaling puts different trading days on a comparable scale, facilitating the analysis of volume trends relative to price movements. High trading volumes can indicate strong interest in the stock, either bullish or bearish, and are often a precursor to major price shifts.

These engineered features enable more nuanced analyses and richer interpretations of the data, serving

Visualization of Engineered Features:

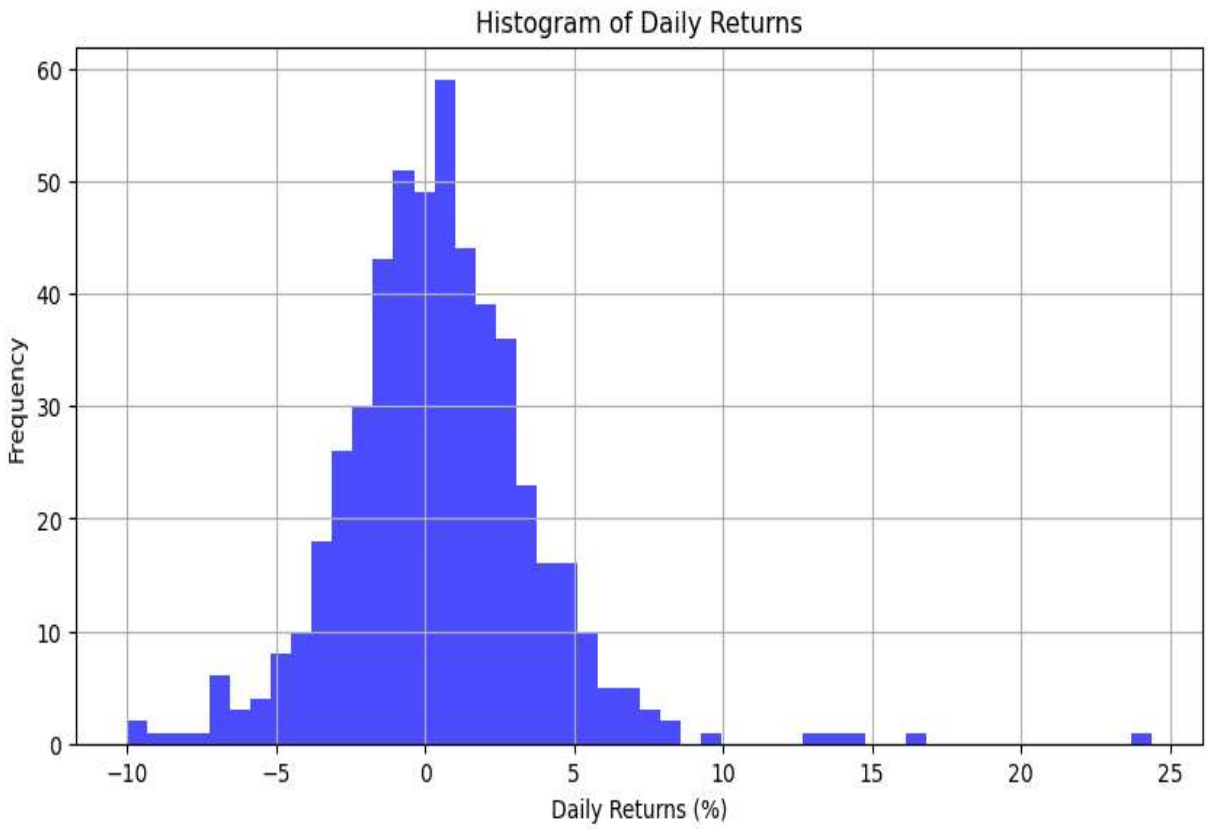
- The overlay plot of the 7-day moving average on the daily closing prices provided insights into market trends versus daily volatility.



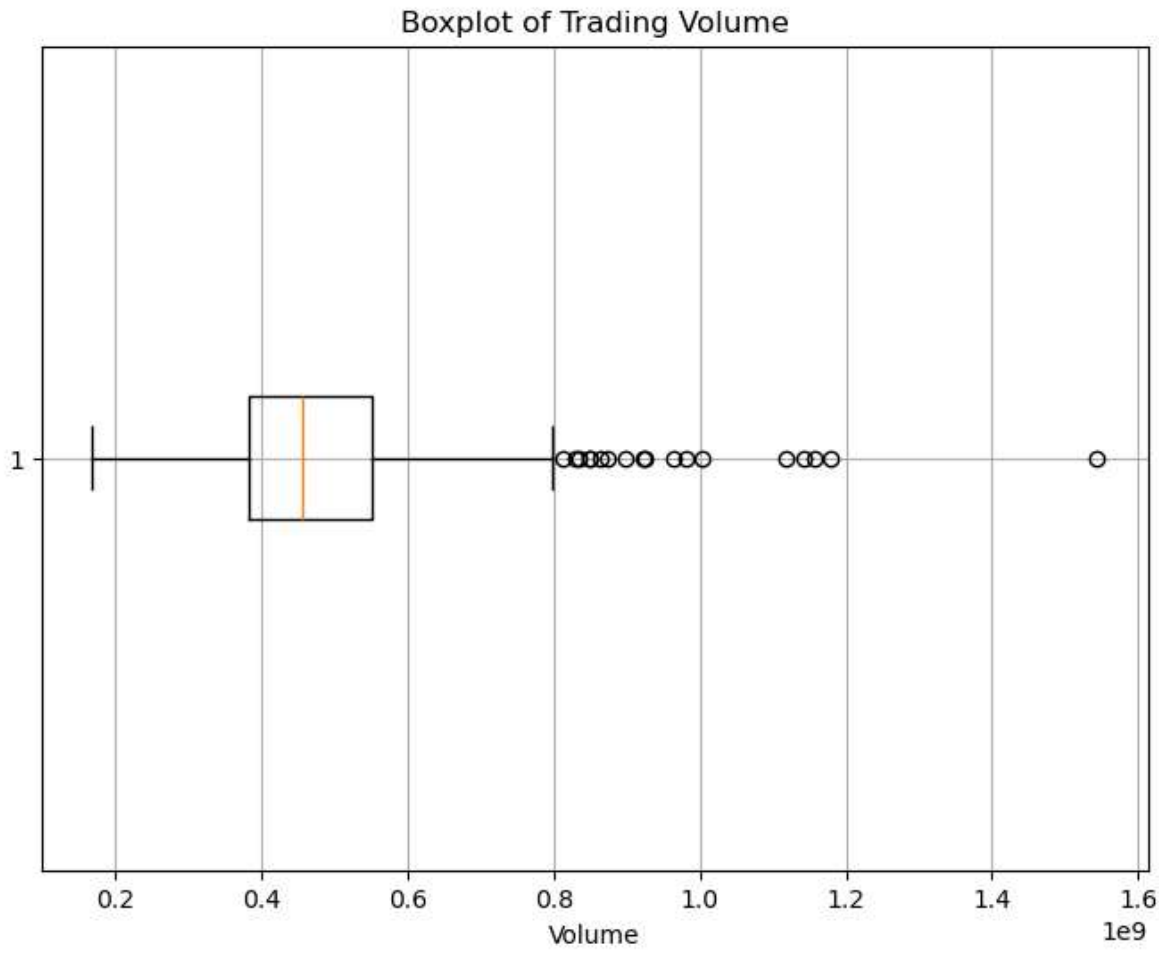
Task 3: Data Visualization

Detailed Analysis of Plots:

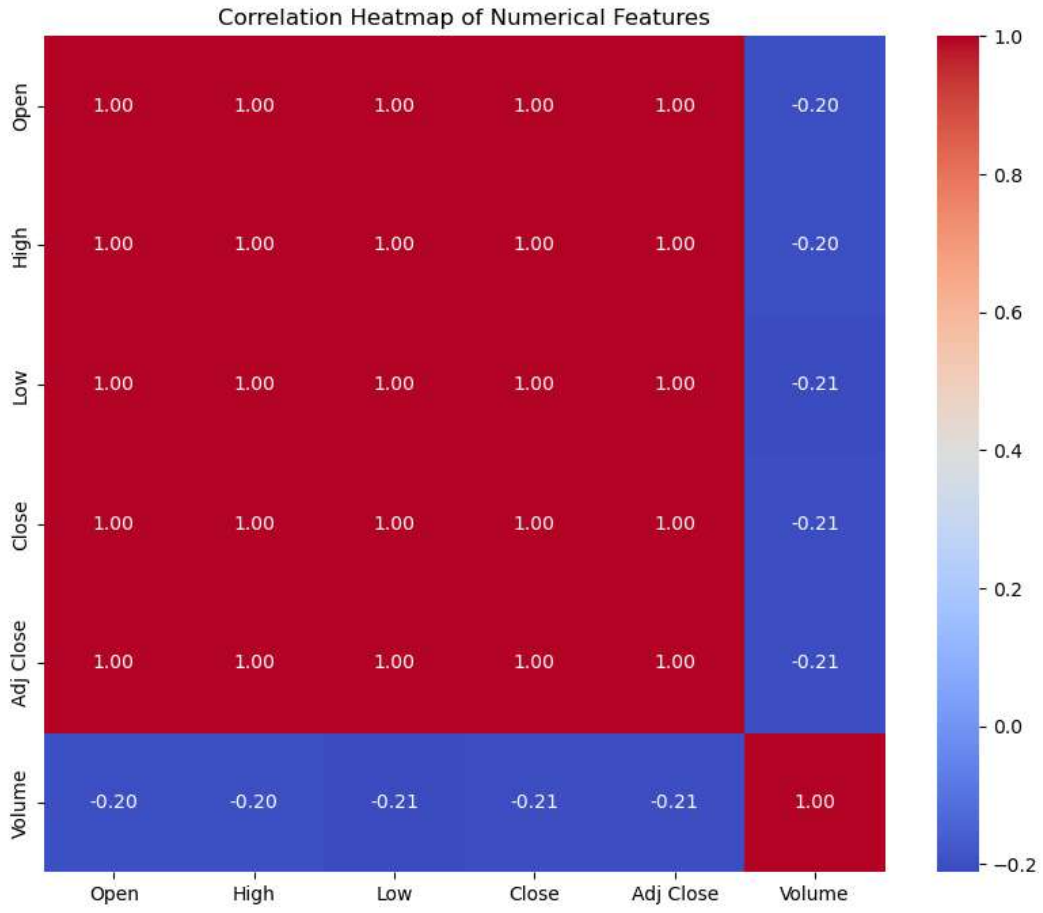
- **Histogram of Daily Returns:**The histogram of daily returns presents a central tendency around zero, suggesting that on most days, the returns fluctuate slightly above or below no change. The distribution appears roughly normal but with heavy tails, indicating occasional days with significantly higher or lower returns. This skewedness towards both ends can be critical for risk assessment, as it implies that while most days are stable, there are rare days with extreme fluctuations that could either mean substantial gains or significant losses. The presence of these outliers underscores the necessity for robust risk management strategies in trading and investment planning.



- **Boxplot of Trading Volume:** The boxplot of trading volumes shows a relatively compact interquartile range, suggesting that typical trading volumes do not vary dramatically most of the time. However, the numerous outliers indicate days with exceptionally high trading volumes, which could correlate with major market events or news impacting Nvidia. These outliers are crucial for identifying days of high investor interest and potential volatility, offering insights into when the market might be more unpredictable or susceptible to external influences.



- **Correlation Heatmap:** The correlation heatmap reveals strong correlations among the Open, High, Low, Close, and Adjusted Close prices, which is expected as these prices are intrinsically linked within each trading day. The negative correlations between these prices and the Volume suggest that on days with higher volumes, the stock price might decrease, indicating selling pressure, or perhaps the higher volumes are due to significant price movements either upward or downward. This inverse relationship provides a basis for further analysis into how trading volumes might affect price stability or be an indicator of major price shifts.



Part 2: Machine Learning

Task 1: Clustering with KMeans

KMeans clustering is utilized to segment stock trading days into distinct groups based on features such as daily returns and trading volumes. This method helps in identifying patterns that are not immediately obvious. Clustering days into similar groups can reveal common behaviors or anomalies in stock movements, which can be crucial for strategic planning in trading and investment.

Methodology

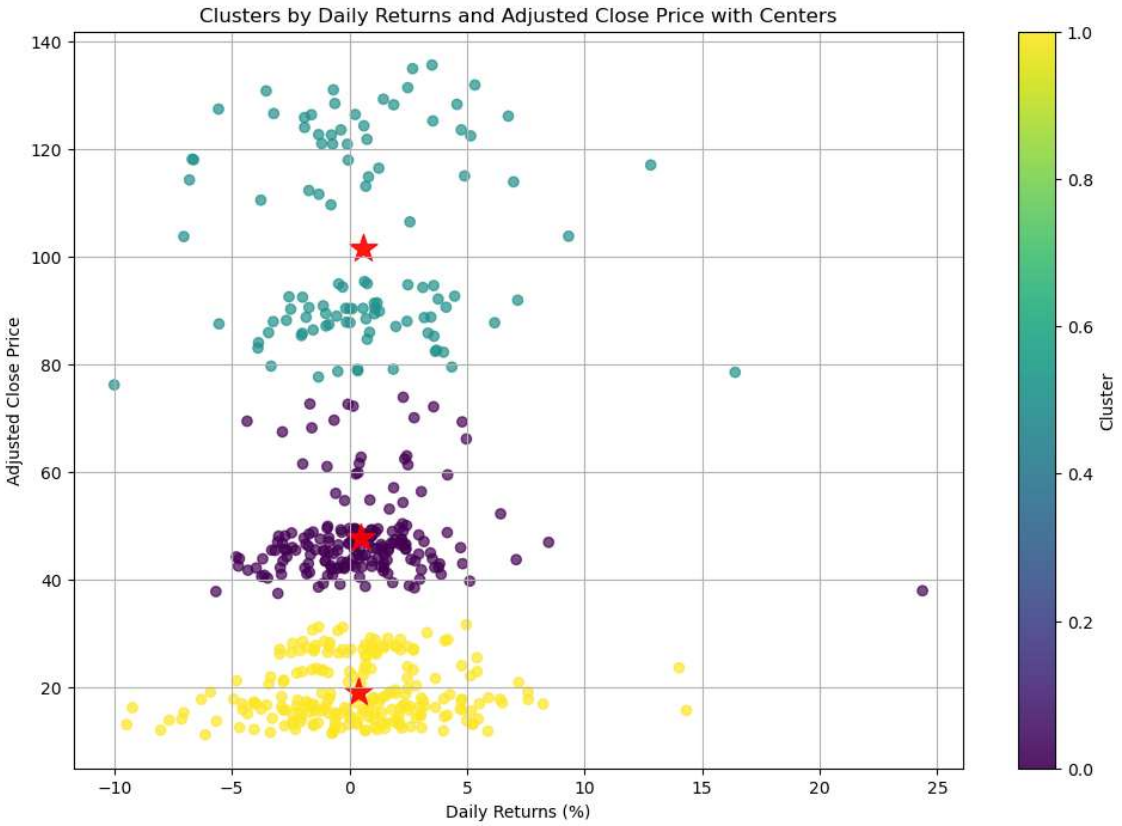
- Feature Selection: Daily returns and trading volumes are chosen because they directly reflect market activity and investor sentiment. These features are likely scaled or normalized before clustering to ensure that one feature doesn't dominate the others due to scale differences.
- Optimal Number of Clusters: The elbow method is typically used to determine the optimal number of clusters by identifying the point where the decrease in the sum of

squared distances (inertia) starts to diminish, indicating that adding more clusters does not significantly improve the model fit.

Interpretation of Results

Clusters Characterization: Each cluster represents a group of days with similar stock price movements and trading behaviors. For example:

- **Cluster 0:** Indicates days with moderate returns, possibly normal trading conditions, and a relatively stable adjusted close price.
- **Cluster 1:** Represent days with high volatility in returns, low trading volume, high adjusted close price.
- **Cluster 2:** Represent days with low volatility in returns and high trading volume, with an low adjusted close price



Task 2: Other machine learning methods

Predictive Modeling:

Linear Regression:

- Predicted future closing prices, evaluated by RMSE, highlighting the model's accuracy.

Model Evaluation

The Linear Regression model, implemented to forecast the closing prices of Nvidia stock, yielded an RMSE of 0.62794816494417. This exceptionally low error metric indicates a high level of accuracy in the predictions made by the model. The small value of RMSE demonstrates that the model's predicted values are very close to the actual data points, suggesting that the model has effectively captured the underlying trends and patterns in the stock price data.

Interpretation of Results

The low RMSE value is indicative of the model's robustness and its capability to generalize well over the data it was tested on. This level of precision in predictive modeling is crucial for applications in financial markets where accuracy in price forecasting can significantly influence investment decisions and risk management strategies. It suggests that the features selected for the model, including the Open, High, Low, and Volume prices, provide substantial information to predict the Close prices accurately.

Logistic Regression:

- Used to predict the likelihood of a stock price increase the next day. The model's effectiveness was quantified through accuracy metrics and a detailed classification report.

Model Performance Metrics

The Logistic Regression model achieved an overall accuracy of 50%, with the following detailed metrics:

- **Precision for Class 0 (stock price will not increase):** 0.00, indicating that the model failed to correctly predict any negative class instances (no price decrease predictions were correct).
- **Recall for Class 0:** 0.00, meaning the model did not correctly identify any of the actual negatives.
- **F1-Score for Class 0:** 0.00, reflecting the poor performance on the negative class due to zero precision and recall.
- **Precision for Class 1 (stock price will increase):** 0.50, meaning that when the model predicts an increase, it is correct 50% of the time.
- **Recall for Class 1:** 1.00, indicating that the model identified all actual increases.
- **F1-Score for Class 1:** 0.67, which is relatively better, showing a moderate balance between precision and recall for the positive class.

Interpretation of Results

- The model's perfect recall for Class 1 suggests it has a strong ability to capture all positive instances (price increases). However, its precision for Class 1 is only 50%, indicating that only half of the predicted price increases were actual increases.
- The zero precision and recall for Class 0 are concerning as they imply the model is incapable of identifying any true negatives, which could be a result of an imbalanced dataset or a model that is biased towards predicting increases.
- The overall accuracy of 50% might not seem impressive, but it's crucial to consider this in the context of the data's distribution and the business or trading strategies it supports. The high recall for Class 1 could be advantageous in scenarios where missing out on predicting a price increase is more costly than falsely predicting an increase.

For both the models the training and the testing set are split in ratio of 4:1

Gradient Boosting Regressor Using Randomized Search

To enhance the predictive accuracy of the Gradient Boosting Regressor model applied to Nvidia's stock price data, we implemented Randomized Search for hyperparameter tuning. This method is an efficient alternative to GridSearchCV, providing a comprehensive approach to exploring optimal combinations of multiple hyperparameters.

Methodology

The Randomized Search was configured to evaluate 10 different combinations of parameters across 3 cross-validation folds, resulting in a total of 30 fits. This approach not only ensures robustness in our model's performance but also maintains computational efficiency. The parameters tuned included:

- **n_estimators:** The number of sequential trees to be modeled. Higher numbers typically offer better performance but increase the risk of overfitting.
- **learning_rate:** Controls the contribution of each tree to the final outcome and helps in reducing overfitting.
- **max_depth:** The maximum depth of each tree. Deeper trees can model more complex patterns but may lead to overfitting.
- **min_samples_split:** The minimum number of samples required to split an internal node.
- **min_samples_leaf:** The minimum number of samples required to be at a leaf node.

Results

The optimal parameters obtained through Randomized Search were:

- **n_estimators:** 300
- **learning_rate:** 0.05
- **max_depth:** 4
- **min_samples_split:** 4
- **min_samples_leaf:** 1

With these settings, the Gradient Boosting Regressor achieved a Root Mean Squared Error (RMSE) of 1.13035 on the test set. This RMSE represents a slight improvement over the baseline model, illustrating the efficacy of the hyperparameter tuning process.

Interpretation

The improvement in RMSE, though modest, signifies that the adjustments in the hyperparameters have positively impacted the model's ability to predict Nvidia's stock prices with greater precision. The increase in the number of estimators and the fine-tuning of the tree's complexity parameters helped capture more subtle patterns in the data without significantly increasing the risk of overfitting.

Conclusion

This report has explored several machine learning and statistical techniques to analyze and predict the stock prices of Nvidia. Utilizing Linear Regression, Logistic Regression, Gradient Boosting Regressor with Randomized Search, and KMeans Clustering, combined with sophisticated data visualization and feature engineering, the study has provided in-depth insights into the dynamics of Nvidia's stock movements.

Summary of Findings:

- Linear Regression yielded the most accurate predictions with an RMSE of 0.62794816494417 and an R-squared of 0.9996738122564774, demonstrating exceptional predictive power and suggesting that the stock price movements are mostly linear and well-explained by the features used.
- Logistic Regression was utilized to classify the direction of stock price movements, achieving a 50% accuracy. Despite its modest performance, it showed a high recall for predicting price increases, indicating its utility in scenarios where failing to predict an increase could result in missed opportunities.
- Gradient Boosting Regressor, optimized via Randomized Search, also showed high predictive accuracy with an RMSE of 1.0915679296037941 and an R-squared of 0.9990143525962691. The model was robust, handling various nonlinear aspects of the data effectively.

- KMeans Clustering provided valuable insights into the behavioral segmentation of trading days, helping to identify patterns and anomalies in trading activities that are not immediately apparent from supervised methods.

Implications for Trading Strategies:

The combination of predictive modeling and clustering offers a robust framework for developing trading strategies. While predictive models provide direct forecasts of stock prices, clustering reveals underlying patterns that can inform risk management and trading decisions. For instance, days clustered with high volatility might indicate a need for more conservative trading strategies, whereas stable days could be opportunities for more aggressive actions.