Siddhi Sunil Nalawade
Siddhisu@buffalo.edu
50613176

## Big-data Processing and Machine Learning

### Introduction

This report analyzes Nvidia's stock prices from July 2022 to July 2024 using data processing and machine learning. It covers data cleaning, visualization, and predictive modeling with K-Means clustering and LSTM. The goal is to identify patterns and improve stock trend predictions.

### Part 1: Big Data Processing

### Data Cleaning and Exploration

### 1. Load Dataset and Display Metadata

- The code imports necessary libraries, loads Nvidia's historical stock price data into a pandas Data Frame, and displays its structure.
- The output reveals **523 rows and 7 columns**, with some missing values in **Open, High, Close, and Volume**. The first five rows show stock price details, confirming that the Date column needs conversion and numerical data is in floating-point format.

First 5 Rows of Dataset:

| | Date | Open | High | Low | Close | Adj Close | Volume |
|---|---|---|---|---|---|---|---|
| **0** | 2022/7/1 | 14.899 | 15.063000 | 14.392 | 14.523 | 14.506663 | 577610000.0 |
| **1** | 2022/7/5 | 14.175 | 14.971000 | 14.055 | 14.964 | 14.947166 | 651397000.0 |
| **2** | 2022/7/6 | 15.010 | 15.319000 | 14.789 | 15.130 | 15.112980 | 529066000.0 |
| **3** | 2022/7/7 | 15.456 | 15.945000 | 15.389 | 15.858 | 15.840160 | 492903000.0 |
| **4** | 2022/7/8 | 15.430 | 16.037001 | 15.389 | 15.838 | 15.820185 | 467972000.0 |

### 2. Check and Handle Missing Values

- Checked missing values using df.isnull().sum(), found missing data in Date, Open, High, Close, and Volume.
- Forward-filled (ffill) missing values in Date, Open, High, and Close to maintain continuity.
- Filled Volume with its median to prevent data skewing.
- After processing, df.isnull().sum() confirmed zero missing values.

### 3. Convert Date to Proper Format

Converted the **Date** column to datetime format and reformatted it to **"DD-MM-YYYY"**, ensuring consistency in date representation.

| | Date | Date_formatted |
|---|---|---|
| 0 | 2022-07-01 | 01-07-2022 |
| 1 | 2022-07-05 | 05-07-2022 |
| 2 | 2022-07-06 | 06-07-2022 |
| 3 | 2022-07-07 | 07-07-2022 |
| 4 | 2022-07-08 | 08-07-2022 |

### 4. Compute Basic Statistics

Computed Min, Max, Mean, Median, and Standard Deviation for key numerical features like Open, High, Low, Close, Adj Close, and Volume.

Below is the output:

Open:

- Min: 10.971
- Max: 139.800003
- Mean: 46.92134795411089
- Median: 42.321999
- Standard Deviation: 32.99460807074795

High:

- Min: 11.735
- Max: 140.759995
- Mean: 47.77337285468452
- Median: 43.0
- Standard Deviation: 33.56795208828847

Low:

- Min: 10.813
- Max: 132.419998
- Mean: 46.00918748183556
- Median: 41.654999
- Standard Deviation: 32.21017457790228

Close:

- Min: 0.0
- Max: 1.0
- Mean: 0.28724432436351643

- Median: 0.2500140768616104
- Standard Deviation: 0.26452802363423594

Adj Close:

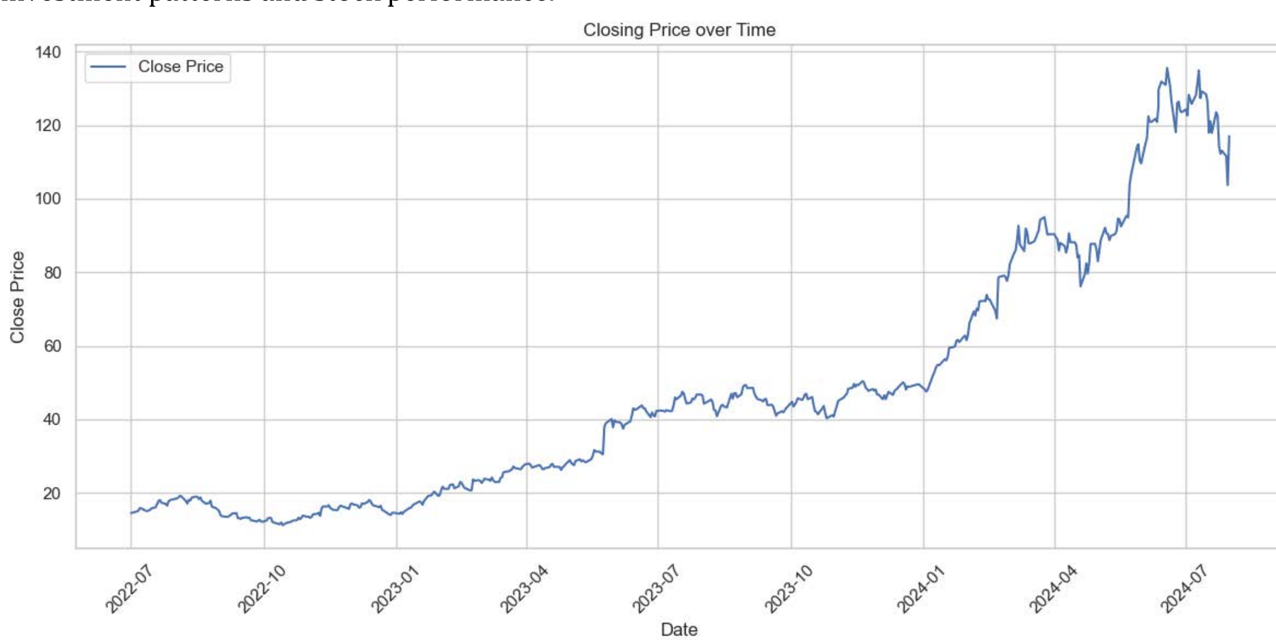
- Min: 11.217702
- Max: 135.580002
- Mean: 46.93218490057361
- Median: 42.304337
- Standard Deviation: 32.892753324228046

Volume:

- Min: 167934000.0
- Max: 1543911000.0
- Mean: 483429610.3250478
- Median: 457328000.0
- Standard Deviation: 157556862.10013533

## 5. Data Visualization

- The stock price remained stable in 2022, gradually increased in 2023, and showed a sharp upward trend in 2024.
- A significant surge in early 2024 was followed by fluctuations, indicating market corrections and volatility.
- This visualization highlights key trends and turning points, useful for analyzing investment patterns and stock performance.



## Feature Engineering

### 1. Calculate Daily Returns and Find Top 10 Days

| | Date | Daily_Return |
|---|---|---|
| 226 | 2023-05-25 | 0.243696 |
| 412 | 2024-02-22 | 0.164009 |
| 92 | 2022-11-10 | 0.143293 |
| 162 | 2023-02-23 | 0.140214 |
| 522 | 2024-07-31 | 0.128121 |
| 476 | 2024-05-23 | 0.093197 |
| 285 | 2023-08-21 | 0.084713 |
| 105 | 2022-11-30 | 0.082379 |
| 17 | 2022-07-27 | 0.076030 |
| 140 | 2023-01-23 | 0.075901 |

### 2. Compute 7-Day Moving Average and Plot

- The 7-day moving average smooths short-term fluctuations, providing a clearer view of the overall trend in stock prices.
- The moving average closely follows the closing price, highlighting long-term growth and filtering out daily volatility.
- This technique helps in identifying trends, potential reversals, and market momentum for better trading decisions.

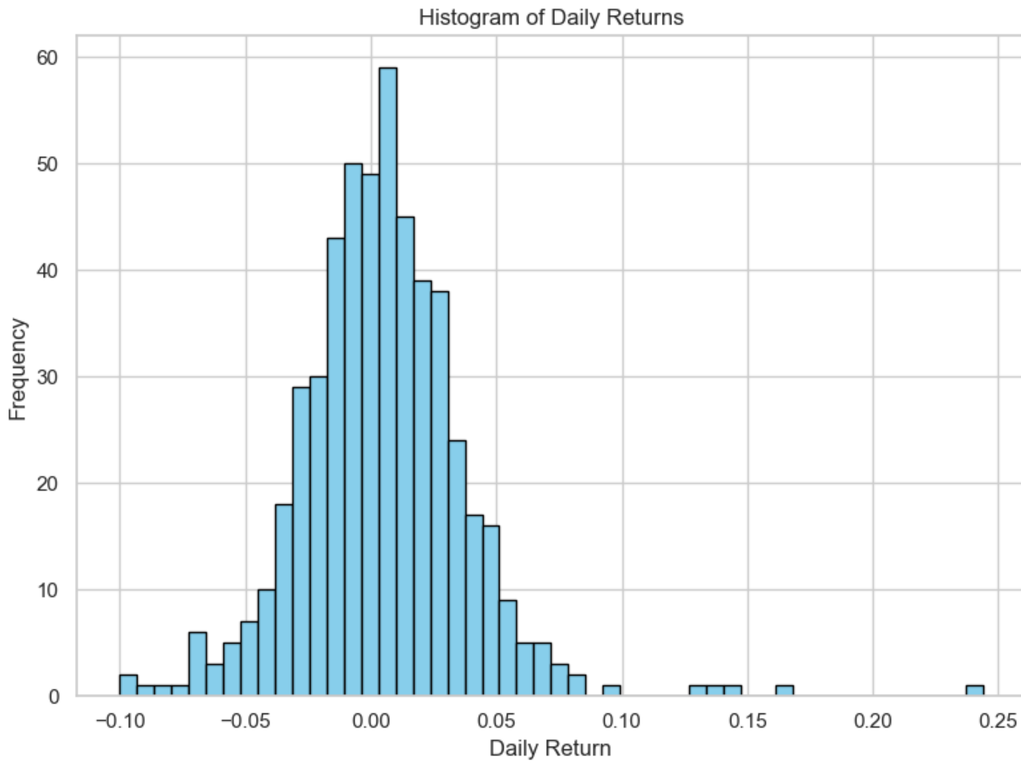
### 3. Normalize Trading Volume and Find Top 10 Days

| Date | Volume | Normalized_Volume |
|---|---|---|
| 226 | 2023-05-25 | 1.543911e+09 1.000000 |
| 43 | 2022-09-01 | 1.178865e+09 0.734701 |
| 288 | 2023-08-24 | 1.156044e+09 0.718115 |
| 423 | 2024-03-08 | 1.142269e+09 0.708104 |
| 162 | 2023-02-23 | 1.117995e+09 0.690463 |
| 229 | 2023-05-31 | 1.002580e+09 0.606584 |
| 25 | 2022-08-08 | 9.818590e+08 0.591525 |
| 264 | 2023-07-21 | 9.637690e+08 0.578378 |
| 289 | 2023-08-25 | 9.253410e+08 0.550450 |
| 228 | 2023-05-30 | 9.234010e+08 0.549040 |

## Data Visualization

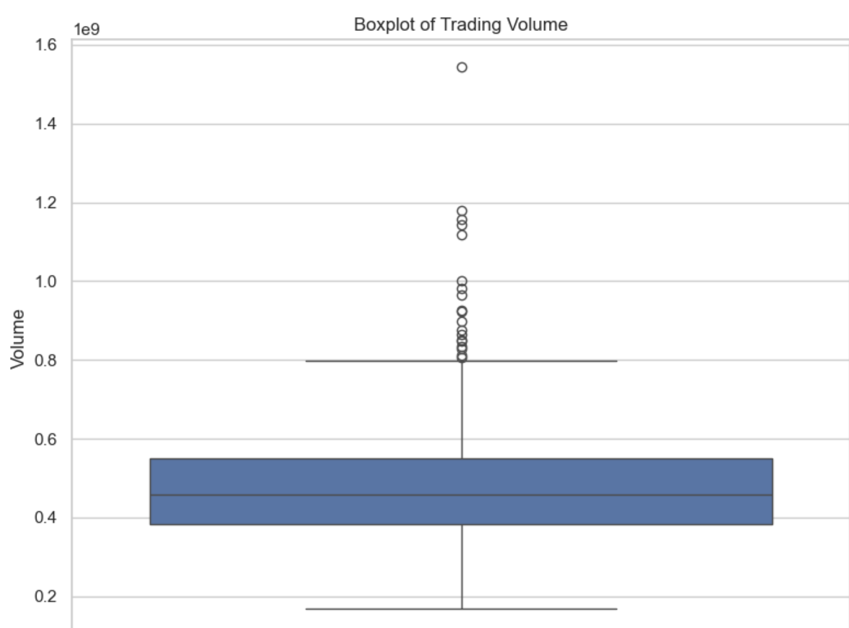
### 1. Histogram of Daily Returns

- The distribution of daily returns is approximately normal, with most returns clustered around 0%, indicating small daily price changes.
- There are some extreme values on both ends, showing occasional large gains or losses, reflecting market volatility and sudden price swings.




## 2. Boxplot of Trading Volume

- The boxplot reveals outliers indicating periods of unusually high trading volume, suggesting market spikes.
- Most trading volumes are concentrated around the median, showing a relatively stable distribution with occasional surges.



### 3. Correlation Heatmap of Numerical Features

- The heatmap shows a strong positive correlation (1.00) among Open, High, Low, Close, and Adjusted Close prices, indicating they move together.
- Trading volume has a weak negative correlation (~ -0.21) with price-related features, suggesting that volume changes do not strongly influence price fluctuations.
- This analysis helps in understanding how different stock features relate to each other, aiding in feature selection for machine learning models.


### Part 2: Machine Learning

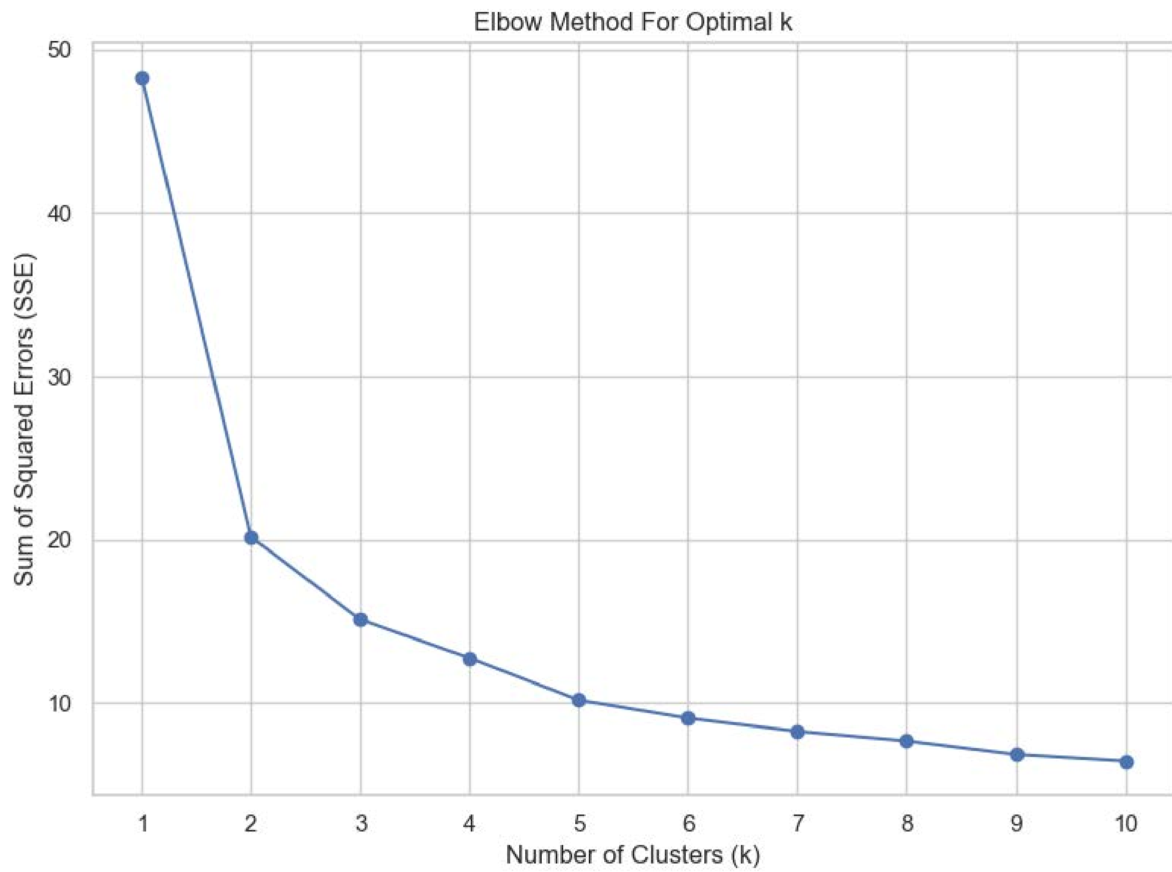
### Clustering with K- Means

### 1. Select Features for Clustering

Normalized  Daily Return, Adjusted Close Price, and Volume using Min-Max Scaling and selected them for K- Means clustering.

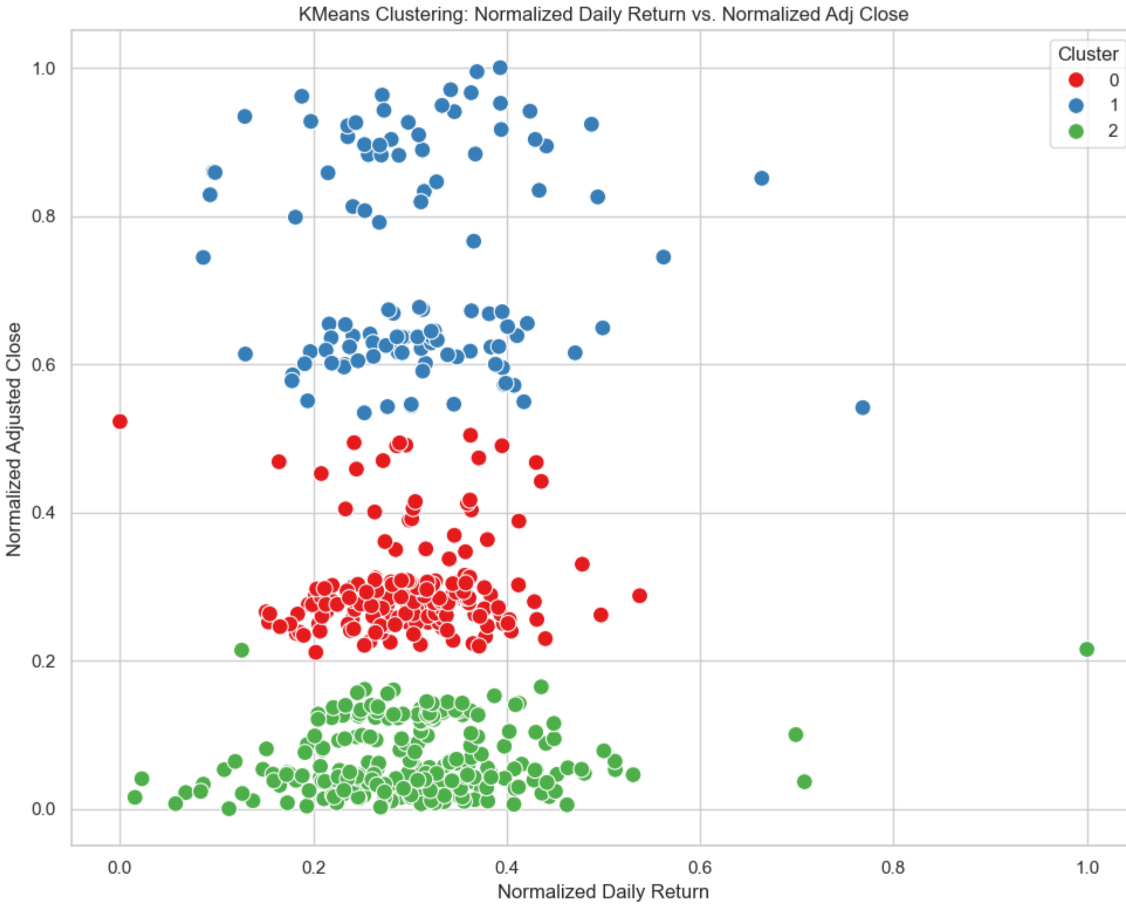
## 2. Find Optimal Clusters with Elbow Method

- Used the Elbow Method by plotting Sum of Squared Errors (SSE) against different values of k (number of clusters).
- The graph shows a **sharp decline at k = 3**, indicating the optimal number of clusters for KMeans.




## 3. Apply K- Means and Visualize Clusters

- Applied KMeans clustering with k = 3, grouping stock behavior based on Daily Return and Adjusted Close Price.
- The scatter plot shows three distinct clusters, representing different market trends and volatility levels.




## 4. Interpret and Analyze Cluster Insights

Cluster Analysis and Insights

The clustering results reveal three distinct groups of stock behavior:

- **Cluster 0:** Represents moderately priced stocks with stable daily returns and lower trading volume. These stocks tend to have minimal fluctuations, making them relatively stable investments.
- **Cluster 1:** Includes high-performing stocks with significantly higher adjusted closing prices and slightly better daily returns. These stocks are likely associated with strong market trends and increased investor confidence.
- **Cluster 2:** Consists of lower-priced stocks with higher volatility. These stocks experience larger fluctuations in daily returns and tend to have more unpredictable movements.

Cluster 0 Statistics:

| | Daily_Return | Normalized_Volume | Adj Close |
|---|---|---|---|
| count | 185.000000 | 185.000000 | 185.000000 |
| mean | 0.003160 | 0.223455 | 48.191037 |
| std | 0.024645 | 0.108947 | 8.217178 |
| min | -0.100046 | 0.022003 | 37.463783 |
| 25% | -0.010686 | 0.154549 | 43.032112 |
| 50% | 0.002970 | 0.195976 | 45.945034 |
| 75% | 0.018804 | 0.263237 | 48.983681 |
| max | 0.084713 | 0.718115 | 76.193741 |

Cluster 1 Statistics:

| | Daily_Return | Normalized_Volume | Adj Close |
|---|---|---|---|
| count | 110.000000 | 110.000000 | 110.000000 |
| mean | 0.006633 | 0.198195 | 102.141636 |
| std | 0.036687 | 0.116575 | 17.873822 |
| min | -0.070436 | 0.004344 | 77.652985 |
| 25% | -0.016006 | 0.110559 | 87.769791 |
| 50% | 0.004675 | 0.190869 | 92.607392 |
| 75% | 0.026720 | 0.255809 | 120.968546 |
| max | 0.164009 | 0.708104 | 135.580002 |

Cluster 2 Statistics:

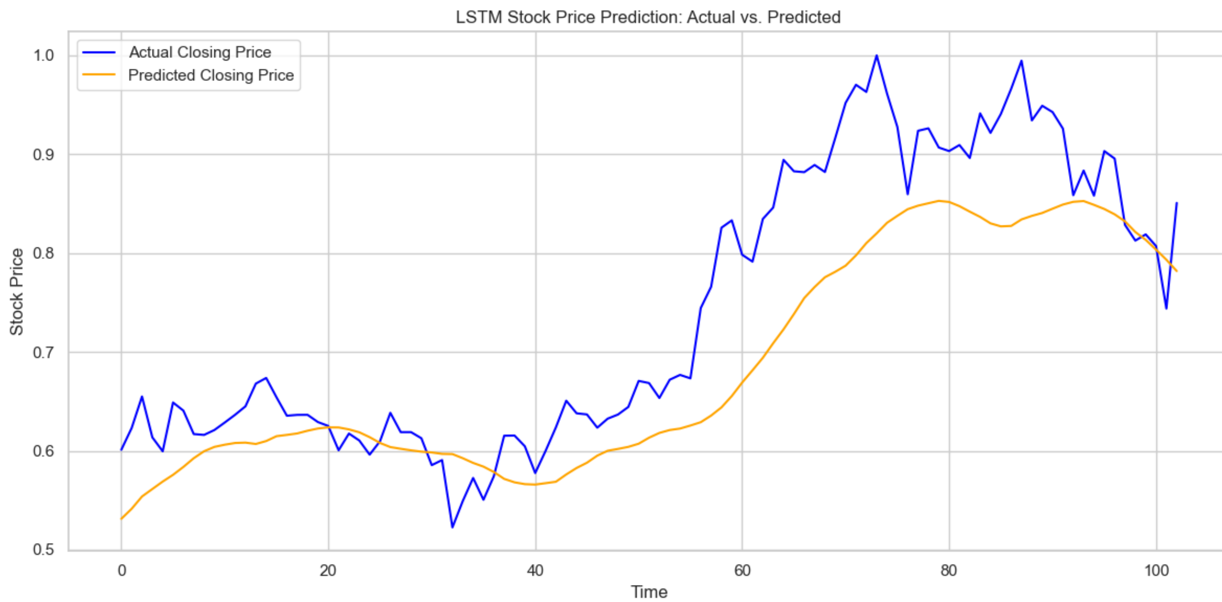
| | Daily_Return | Normalized_Volume | Adj Close |
|---|---|---|---|
| count | 228.000000 | 228.000000 | 228.000000 |
| mean | 0.004660 | 0.249023 | 19.274609 |
| std | 0.037779 | 0.114528 | 5.760822 |
| min | -0.094726 | 0.000000 | 11.217702 |
| 25% | -0.016239 | 0.175659 | 14.823009 |
| 50% | 0.005115 | 0.230611 | 17.333285 |
| 75% | 0.023597 | 0.289769 | 23.500787 |
| max | 0.243696 | 1.000000 | 37.964703 |

## Other machine learning methods

## Train a regression model LSTM to predict the closing price of the stock based on historical data

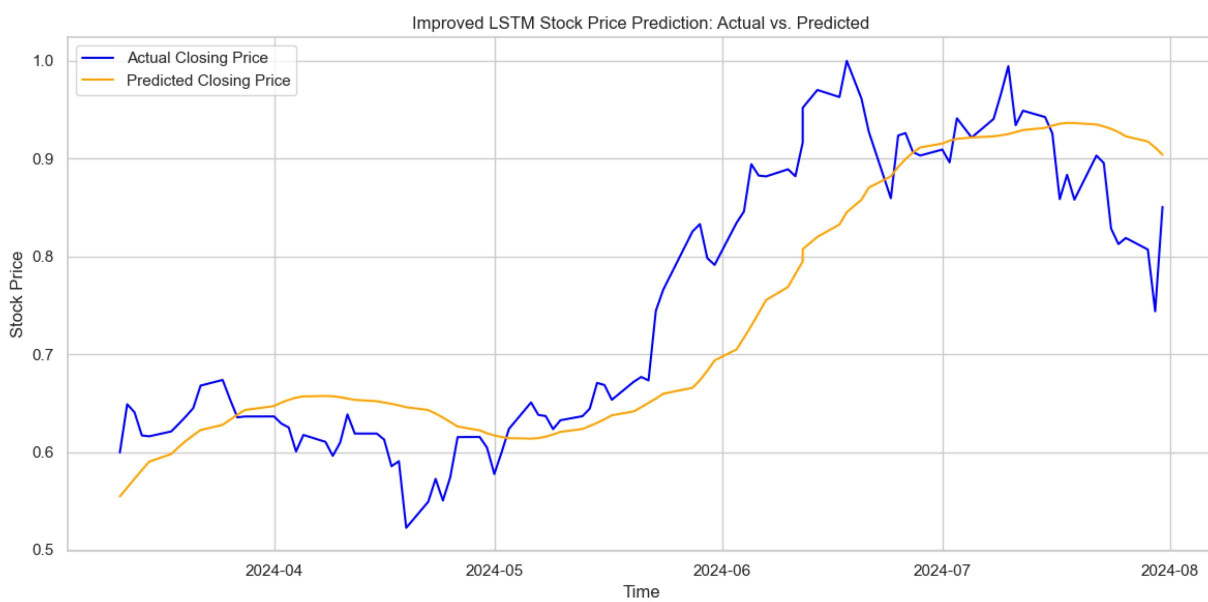
## 1. Stock Price Prediction - LSTM

- Trained an LSTM model on historical stock prices, achieving an MSE of 0.0065, indicating reasonable accuracy.
- The plot shows that the model captures overall stock trends but smooths short-term fluctuations.




## LSTM Model with Hyperparameter Tuning

- Enhanced the LSTM model with more layers and dropout regularization, reducing overfitting.
- Achieved MSE = 0.0054 and $R^2$ = 0.7285, showing improved prediction accuracy while capturing market trends more effectively.



## 2. Trend Classification

## Trend Classification: Implement a classification model Support Vector Machine to predict whether the stock price will go up or down the next day.

- Implemented an SVM model to predict whether the stock price will go up or down the next day based on Open, High, Low, and Volume features.
- Achieved an accuracy of 48.57%, indicating that the model struggles to make reliable predictions, likely due to market volatility and feature limitations.

### SVM Model with Hyperparameter Tuning

- Tuned the SVM model using GridSearchCV with different values of C and gamma to improve prediction accuracy.
- Achieved a best cross-validation accuracy of 57.10% and a test set accuracy of 52.38%, showing slight improvement but still limited by stock market volatility.

- **Conclusion:**
  K- Means clustering successfully categorized stocks into three groups based on volatility and performance.
- LSTM provided the best stock price predictions, with MSE = 0.0054 and $R^2$ = 0.7285, capturing market trends effectively.
- SVM struggled with trend classification (52.38% accuracy), highlighting the difficulty of short-term stock movement predictions.
- Overall, machine learning techniques provided valuable insights, with LSTM excelling in forecasting and clustering helping in market segmentation.