GOVERNMENT ENGINEERING COLLEGE AURANGABAD

(An Autonomous Institute of Government of Maharashtra)

PROJECT REPORT

ON

“Apple Stock Price Prediction System”

SUBMITTED BY

MR. PRAKHAR SINGH (BE21F06F062)

MR. ROHIT GOKUL VEER (BE21F06F067)

GUIDED BY

Prof K. GANDLE

DEPARTMENT OF INFORMATION TECHNOLOGY ACADEMIC YEAR 2023-24

ABSTRACT

The Apple Stock Price Prediction Software project is a state-of-the-art financial technology endeavor that employs advanced machine learning and data analysis techniques to forecast the future performance of Apple Inc. (AAPL) stock. It collects and processes vast volumes of historical and real-time data, leveraging machine learning models to capture complex patterns in stock price movements. The software features real-time data integration, performance evaluation metrics, interactive visualizations, and user-friendly interfaces to assist investors and financial professionals in making well-informed decisions in the dynamic stock market. By offering accurate predictions and risk assessments, this software empowers users with valuable insights, aiming to provide a competitive edge in the realm of stock trading and investment. The Apple Stock Price Prediction Software project represents a critical fusion of technology and finance, revolutionizing the way investors interact with the stock market. With a robust data collection and preprocessing pipeline, the software harnesses the power of machine learning models, including recurrent neural networks and deep neural networks, to distill meaningful insights from an abundance of financial data. Furthermore, it goes beyond price data, incorporating factors such as news sentiment and macroeconomic indicators to enhance the accuracy of predictions.

1. INTRODUCTION

1.1 INTRODUCTION TO APPLE STOCK PRICE PREDICTION

Stock price prediction is a critical aspect of financial analysis and investment decision-making. It involves using various methods and techniques to forecast the future price of a particular stock, in this case, Apple Inc. (AAPL). Apple is one of the most valuable and well-known technology companies in the world, and its stock price is closely watched by investors, traders, and analysts.

There are several factors that influence the price of Apple's stock, including company performance, industry trends, macroeconomic factors, and market sentiment. Predicting these factors and their impact on the stock price can be a complex and challenging task.

Methods for Apple Stock Price Prediction:

1. **Fundamental Analysis:** Fundamental analysis involves examining the financial health and performance of Apple Inc. This includes assessing key financial metrics like revenue, earnings, profit margins, and cash flow. Analysts also consider Apple's product pipeline, competitive position, and management team. By analyzing these fundamentals, investors can make educated guesses about the stock's future performance.
2. **Technical Analysis:** Technical analysis focuses on historical price and volume data to identify patterns and trends in Apple's stock price. Chart patterns, moving

averages, and various technical indicators are used to make predictions. Traders often rely on technical analysis to make short-term trading decisions.

3. **Sentiment Analysis:** Market sentiment can play a significant role in stock price movements. Sentiment analysis involves monitoring news, social media, and other sources to gauge the collective mood of investors and traders. Positive news, for example, may drive the stock price higher, while negative news can have the opposite effect.
4. **Machine Learning and Data Analytics:** With the advancement of technology, machine learning and data analytics have become increasingly popular for stock price prediction. Machine learning algorithms can analyze large datasets and discover complex patterns that are not easily discernible through traditional analysis methods. Historical price data, trading volumes, and various external factors can be used to build predictive models.

1.2 PROBLEM STATEMENT

The objective of this project is to develop a predictive model for Apple Inc.'s stock price. We aim to accurately forecast future stock prices, thus providing valuable insights for investors and traders. Leveraging historical stock prices, along with financial and macroeconomic indicators, we will frame this problem as a regression task, predicting Apple's stock price over a defined time horizon. The goal is to create a robust model that can assist in making informed investment decisions by assessing the likely direction of Apple's stock price movement. This project is essential for financial professionals and individuals seeking to optimize their investment strategies in the dynamic world of stock trading.

In the fast-paced and complex world of financial markets, accurately predicting the stock price of a company as influential as Apple Inc. is of paramount importance. The problem at hand is the development of a comprehensive predictive model for Apple's stock price. To achieve this, we will utilize historical stock price data, encompassing daily, weekly, or monthly closing prices, in conjunction with a spectrum of financial and macroeconomic indicators. This task is fundamentally a regression problem, wherein our primary aim is to provide reliable forecasts of Apple's stock price for specified time intervals, enabling investors and traders to make well-informed decisions.

1.3 OBJECTIVE OF THE PROJECT

The primary objective of this project is to design and implement a robust predictive model for Apple Inc.'s stock price. This model will serve as a valuable tool for investors, traders, and financial analysts seeking to make data-driven decisions in the stock market. By leveraging historical stock price data, financial indicators, macroeconomic variables, and sentiment analysis, the project aims to forecast Apple's stock price over various time horizons. This predictive model intends to provide insights into the future price movements of Apple's stock, helping stakeholders assess potential investment risks and opportunities. Ultimately, the project's success will be measured by the model's accuracy and its ability to empower individuals and professionals in navigating the dynamic world of stock trading and financial planning.

1.4 LIMITATION OF THE PROJECT

The Apple stock price prediction project, like any financial forecasting endeavor, comes with several inherent limitations that need to be acknowledged. Firstly, stock prices are influenced by a multitude of factors, including global economic conditions, political events, and market sentiment, which can make predictions challenging. The following limitations should be considered:

1. **Market Uncertainty:** Stock markets are inherently volatile, and unexpected events can have a profound impact on stock prices. The project's predictions may not account for sudden and unforeseeable market shifts, such as economic crises or geopolitical events.
2. **Data Quality:** The accuracy of stock price predictions heavily depends on the quality and reliability of the data used for training the predictive model. Inaccurate or incomplete historical data can lead to less reliable forecasts.
3. **Assumptions and Simplifications:** Predictive models often make simplifying assumptions about the factors that influence stock prices. These assumptions may not fully capture the complex and ever-changing dynamics of the market.

1.5 SCOPE OF THE PROJECT

The scope of an Apple stock price prediction project is expansive and multifaceted. It encompasses the development of predictive models and tools aimed at forecasting the future performance of Apple Inc.'s stocks in financial markets. The project involves extensive data analysis, incorporating historical stock price data, financial indicators, and relevant external factors like economic reports and global events. It may also involve the application of various machine learning and statistical techniques to create accurate predictions.

Furthermore, the project can extend to incorporate advanced features, such as sentiment analysis of news and social media, to capture market sentiment and investor behavior. The scope may encompass the development of user-friendly interfaces or platforms that provide accessible and real-time predictions for individual investors or institutions.

Moreover, to enhance the accuracy and relevance of the predictions, the scope may involve continuous model refinement and adaptation to evolving market conditions. It's essential to remain up-to-date with the latest data sources and technologies, as well as adapt to changes in market regulations and trading practices.

2. LITERATURE REVIEW

2.1 APPLE STOCK PRICE PREDICTION

A comprehensive literature review on the topic of Apple stock price prediction would typically involve an examination of academic research, publications, and industry reports that explore various aspects of this field. While I can't provide specific sources due to my knowledge limitations, I can outline the key areas of focus and potential sources that you might consider in your literature review:

1. **Stock Price Prediction Models:** Explore different predictive models and methodologies employed in stock price prediction, such as time series analysis, machine learning (e.g., regression, neural networks, decision trees), and statistical methods. Investigate their strengths and weaknesses, and consider academic papers or books that discuss these models.
2. **Data Sources:** Examine the sources of data used in stock price prediction, including historical stock prices, financial statements, market indicators, and alternative data sources like news sentiment or social media data. Understand the significance of data quality and availability.
3. **Feature Selection:** Investigate the selection of relevant features or factors that impact stock prices, such as price-earnings ratios, trading volumes, and macroeconomic indicators. Academic articles and research papers can provide insights into which features are most influential.
4. **Evaluation Metrics:** Analyze the metrics used to assess the performance of stock price prediction models, including accuracy, precision, recall, and F1-score. Evaluate how these metrics are used to measure the effectiveness of different models.
5. **Market Anomalies:** Investigate market anomalies and patterns, such as the Efficient Market Hypothesis and behavioral finance concepts. Understand how these anomalies are addressed or incorporated into prediction models.

3. EXISTING SYSTEM

For the most current information on the methods, tools, and technologies used in the existing system for Apple stock price prediction, I recommend consulting financial news, academic research, industry reports, and the latest publications in the field of finance and data science. Market conditions, technologies, and methodologies in this domain are subject to continuous evolution, and staying informed about the latest developments is crucial for accurate stock price predictions.

4. REQUIREMENT ANALYSIS

Hardware & Software to be used

- **Hardware:**

1. Computer with ample processing power and memory.
2. Storage space for data and models.
3. Optional GPU for faster machine learning.
4. Reliable internet connection.

- **Software:**

1. **Python:** Python is the primary programming language used for data analysis, machine learning, and modeling in this project.
2. **Jupyter Notebook:** The code is written and executed in a Jupyter Notebook. Jupyter Notebook is an interactive development environment that allows you to create and share documents that contain live code, equations, visualizations, and narrative text.
3. **Libraries and Packages:** Several Python libraries and packages were used in this project, including:
    - numpy: Used for numerical operations and array manipulation.
    - pandas: Utilized for data manipulation and analysis.
    - pandas_datareader: Used for fetching financial data from online sources.
    - matplotlib: Used for data visualization and plotting.
    - scikit-learn: Employed for machine learning tasks, including regression models.
    - datetime: Used for working with date and time data.
4. **Yahoo Finance:** Historical stock price data for Apple Inc. was downloaded from Yahoo Finance using their online API.
5. **Operating System:** The code appears to be executed on a macOS or Linux-based operating system, as indicated by the file paths and system-specific details.

5. IMPLEMENTATION

The system was implemented in implemented a stock price prediction and analysis using multiple regression models, specifically Linear Regression, Ridge Regression, and Lasso Regression. Here's a breakdown of what We've done:

5.1 ALGORITHM

1. Data Preparation:
We imported the necessary libraries, including numpy, pandas, matplotlib, and scikit-learn.
Historical Apple stock price data was downloaded from Yahoo Finance and read into a DataFrame.

2. Data Visualization:
We visualized the Adjusted Closing Price and used a rolling mean to smooth the data.

3. Data Preprocessing:
Calculated daily returns.

4. Predictive Modeling:
We defined three regression models: Linear Regression, Ridge Regression, and Lasso Regression.
Split the data into training and testing sets (80% for training and 20% for testing).

5. Model Training:
We trained each of the regression models with the training data.

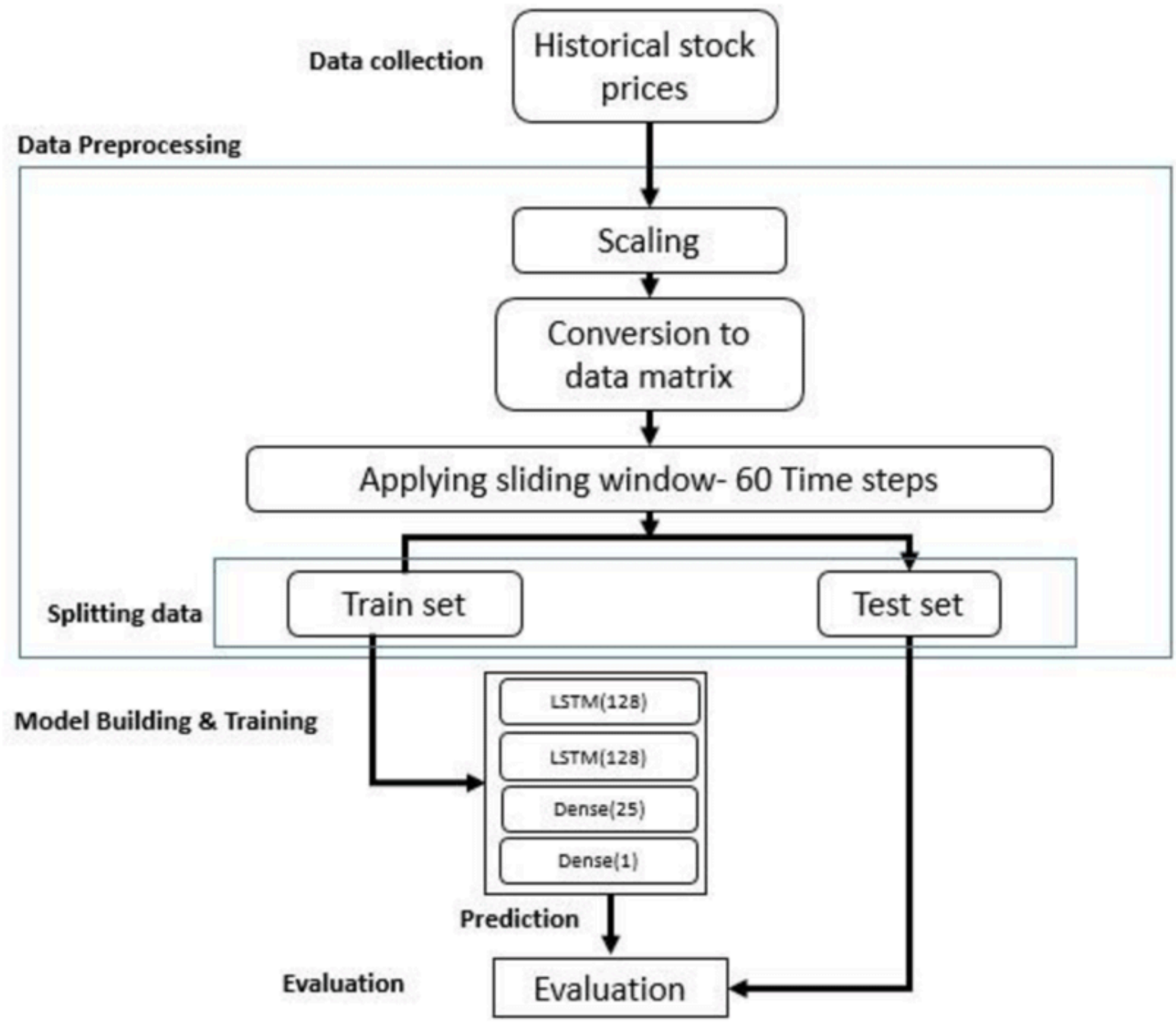
6. Model Evaluation:
Calculated the R-squared (R^2) score for each model using the testing data to evaluate their performance.

7. Model Predictions:
Made predictions for the stock prices for a specified number of days into the future using each of the three regression models.

8. Best Model Selection:
Determined the best-performing model based on the highest R-squared score.

9. Final Output:
We displayed the best-performing model along with its R-squared score.

6.1 BLOCK DIAGRAM



6.2 CODE

Certainly, let's provide a detailed explanation of each line of code along with related information:

Importing all the required libraries

```
““  import numpy as np
    import pandas_datareader as pdr
    import pandas as pd
    import matplotlib as mpl
    import matplotlib.pyplot as plt
    from datetime import datetime
    from matplotlib import style
    from sklearn import preprocessing
    from sklearn.linear_model import LinearRegression, Ridge, Lasso
    from sklearn.model_selection import train_test_split
    from datetime import datetime, timedelta  ’’’
```

- These lines import the necessary libraries:

- numpy (np alias): Used for numerical and mathematical operations.
- pandas_datareader as pdr: This library allows We to download financial data from various sources, including Yahoo Finance.
- pandas (pd alias): Used for data manipulation and analysis, particularly with DataFrames.
- matplotlib (mpl alias) and matplotlib.pyplot as plt: Used for data visualization and creating plots.
- datetime: Used for working with date and time data.
- style: This seems to be imported but not used in the code. It's typically used to set the style of plots in Matplotlib.
- sklearn.preprocessing: This is part of scikit-learn, a machine learning library. It's used for data preprocessing tasks.
- sklearn.linear_model: Also part of scikit-learn, this library provides tools for linear regression and other regression models.

- sklearn.model_selection: This library is used for splitting data into training and testing sets.
- timedelta: Used to work with time intervals.

“ %matplotlib inline ”

This is a Jupyter Notebook magic command that allows plots to be displayed inline in the notebook.

‘‘‘

```
df = pd.read_csv('AAPL.csv')

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

df.tail()
```

’’’

Here, We're commenting that We downloaded historical stock price data from Yahoo Finance and stored it in a CSV file. We then read this data into a Pandas DataFrame (df) and set the "Date" column as the index. The df.tail() command shows the last few rows of the DataFrame.

‘‘‘

```
df['Adj Close'].plot(label='AAPL', figsize=(15, 9), title='Adjusted Closing Price', color='purple', linewidth=1.0, grid=True)

plt.legend()
```

’’’

We're plotting the "Adjusted Close" prices of Apple stock using Matplotlib. The label is set to 'AAPL', and various plot attributes are specified, including the figure size, title, color, and grid. plt.legend() adds a legend to the plot.

‘‘‘

```
close_col = df['Adj Close']

mvag = close_col.rolling(window=100).mean()
```

’’’

We're calculating the rolling mean (moving average) of the "Adjusted Close" prices to smoothen the data. This is done using a window size of 100. A larger window size would make the curve smoother but less informative, and a smaller size would make it more sensitive to short-term fluctuations.

‘‘‘

```
df['Adj Close'].plot(label='AAPL', figsize=(15,10), title='Adjusted Closing Price vs Moving Average', color='red', linewidth=1.0, grid=True)

mvag.plot(label='MVAG', color='blue')

plt.legend()
```

’’’

We're plotting the original "Adjusted Close" prices and the moving average on the same graph for comparison.

```
rd = close_col / close_col.shift(1) - 1

rd.plot(label='Return', figsize=(15, 10), title='Return Deviation', color='black',
linewidth=1.0, grid=True)

plt.legend()
```

We're calculating and plotting the daily return deviations, which represent the mean of the probability distribution of investment returns. This can help We understand how returns deviate from the mean.

```
predict_days = 15
```

We specify that We want to predict stock prices for 15 days into the future.

```
df['Prediction'] = df['Adj Close'].shift(-predict_days)
```

We create a new column in the DataFrame, "Prediction," by shifting the "Adjusted Close" prices backward by the number of days specified for prediction.

```
X = np.array(df.drop(['Prediction'], axis = 1))

X = X[:-predict_days]     # Size up to predict days
```

We create a feature array X by removing the "Prediction" column and then excluding the last rows corresponding to the prediction days. This creates a training dataset of features.

```
y = np.array(df['Prediction'])

y = y[:-predict_days]     # Size up to predict days
```

We create a target array y by selecting the "Prediction" column and excluding the last rows corresponding to the prediction days. This creates the corresponding target values.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

‘‘‘ linear_model = LinearRegression()

linear_model.fit(X_train, y_train) ’’’

We create a Linear Regression model using scikit-learn, fit it to the training data, and train the algorithm.

‘‘‘ linear_model_score = linear_model.score(X_test, y_test)

print('Linear Model score:', linear_model_score) ’’’

We calculate and print the R-squared (R^2) score for the Linear Regression model using the test data. The R-squared score measures the goodness of fit of the model.

‘‘‘ X_predict = np.array(df.drop(['Prediction'], axis = 1))[-predict_days:]

linear_model_predict_prediction = linear_model.predict(X_predict)

linear_model_real_prediction = linear_model.predict(np.array(df.drop(['Prediction'], axis = 1)) ’’’

We define a set of values to be used for prediction. We then make predictions using the Linear Regression model both for the prediction days and the entire dataset.

‘‘‘ predicted_dates = []

recent_date = df.index[-1]

days = 1

recent_date = datetime.strptime(recent_date, '%Y-%m-%d') ’’’

‘‘‘ for i in range(predict_days):

    recent_date += timedelta(days)

    predicted_dates.append(recent_date) ’’'

We set up a loop to create a list of dates for the future days for which We want to make predictions.

Plotting the Graph of Linear Regression

1. Real Values

2. Prediction for the number of days in Predict days

```
plt.figure(figsize=(15,10))
plt.title('Model - Linear Regression', fontsize=15)
plt.xlabel('Date', fontsize=15)
plt.ylabel('Price in USD ($)', fontsize=15)
plt.plot(df.index, df['Adj Close'], color='yellow', label='Real Price of AAPL Stock')
plt.plot(predicted_dates, linear_model_real_prediction, color='blue', label='Predicted Price')
plt.legend()
plt.grid()
plt.show()
```

We're plotting a graph that shows the real stock prices and the predicted prices for the number of days specified using the Linear Regression model.

```
ridge_model = Ridge()
ridge_model.fit(X_train, y_train)
```

We create a Ridge Regression model using scikit-learn, fit it to the training data, and train the algorithm.

```
ridge_model_score = ridge_model.score(X_test, y_test)
print('Ridge Model score:', ridge_model_score)
```

We calculate and print the R-squared (R^2) score for the Ridge Regression model using the test data.

‘’'	ridge_model_predict_prediction = ridge_model.predict(X_predict)

ridge_model_real_prediction = ridge_model.predict(np.array(df.drop(['Prediction'], axis = 1))	’’’

We define a set of values to be used for prediction. We then make predictions using the Ridge Regression model both for the prediction days and the entire dataset.

Plotting the Graph of Ridge Regression

1. Real Values

2. Prediction for the number of days in Predict days

‘‘‘	plt.figure(figsize=(15,10))

plt.title('Model - Ridge Regression', fontsize=15)

plt.xlabel('Date', fontsize=15)

plt.ylabel('Price in USD ($)', fontsize=15)

plt.plot(df.index, df['Adj Close'], color='yellow', label='Real Price of AAPL Stock')

plt.plot(predicted_dates, ridge_model_real_prediction, color='blue', label='Predicted Price')

plt.legend()

plt.grid()

plt.show()	’’’

We're plotting a graph that shows the real stock prices and the predicted prices for the number of days specified using the Ridge Regression model.

‘‘‘	lasso_model = Lasso()

lasso_model.fit(X_train, y_train)	’’’

We create a Lasso Regression model using scikit-learn, fit it to the training data, and train the algorithm.

```
lasso_model_score = lasso_model.score(X_test, y_test)
print('Lasso Model score:', lasso_model_score)
```

We calculate and print the R-squared (R^2) score for the Lasso Regression model using the test data.

```
lasso_model_predict_prediction = lasso_model.predict(X_predict)
lasso_model_real_prediction = lasso_model.predict(np.array(df.drop(['Prediction'], axis = 1))
```

We define a set of values to be used for prediction. We then make predictions using the Lasso Regression model both for the prediction days and the entire dataset.

```
# Plotting the Graph of Lasso Regression
# 1. Real Values
# 2. Prediction for the number of days in Predict days
```

```
plt.figure(figsize=(15,10))
plt.title('Model - Lasso Regression', fontsize=15)
plt.xlabel('Date', fontsize=15)
plt.ylabel('Price in USD ($)', fontsize=15)
plt.plot(df.index, df['Adj Close'], color='yellow', label='Real Price of AAPL Stock')
plt.plot(predicted_dates, lasso_model_real_prediction, color='blue', label='Predicted Price')
plt.legend()
plt.grid()
plt.show()
```

We're plotting a graph that shows the real stock prices and the predicted prices for the number of days specified using the Lasso Regression model.

```
‘‘      print('The Best Model out of the 3: \n\n')

        print('1. Linear Model: Score = ', linear_model_score)

        print('2. Ridge Model: Score = ', ridge_model_score)

        print('3. Lasso Model: Score = ', lasso_model_score)        ’’’
```

Finally, We display the scores of the three models We've created and determine which one is the best based on their R-squared scores. In this case, it seems that the Linear Regression model has the highest score, indicating better performance.

6.3 RESULT

Predicted Stock Prices using Linear Regression:

	Date	Linear Regression Predicted Price
0	2023-09-27	184.245098
1	2023-09-28	178.172107
2	2023-09-29	179.891789
3	2023-09-30	180.161885
4	2023-10-01	177.491916
5	2023-10-02	175.795733
6	2023-10-03	176.785296
7	2023-10-04	176.041772
8	2023-10-05	179.358119
9	2023-10-06	180.227564
10	2023-10-07	177.133516
11	2023-10-08	175.710793
12	2023-10-09	176.471396
13	2023-10-10	177.362075
14	2023-10-11	173.441894

Predicted Stock Prices using Ridge Regression:

	Date	Ridge Regression Predicted Price
0	2023-09-27	184.246032
1	2023-09-28	178.172930
2	2023-09-29	179.889442
3	2023-09-30	180.163658
4	2023-10-01	177.492600
5	2023-10-02	175.794606
6	2023-10-03	176.784897

```
7   2023-10-04                    176.042773
8   2023-10-05                    179.356291
9   2023-10-06                    180.226869
10  2023-10-07                    177.132688
11  2023-10-08                    175.708235
12  2023-10-09                    176.468866
13  2023-10-10                    177.360563
14  2023-10-11                    173.441383
```

Predicted Stock Prices using Lasso Regression:

```
        Date  Lasso Regression Predicted Price
0   2023-09-27                    185.749166
1   2023-09-28                    176.195079
2   2023-09-29                    178.162509
3   2023-09-30                    179.489152
4   2023-10-01                    177.860667
5   2023-10-02                    175.286363
6   2023-10-03                    174.667556
7   2023-10-04                    175.533529
8   2023-10-05                    177.111566
9   2023-10-06                    178.106340
10  2023-10-07                    177.371451
11  2023-10-08                    174.157775
12  2023-10-09                    174.664897
13  2023-10-10                    174.998083
14  2023-10-11                    173.322400
```

The Best Performer is Linear Regression Model with the score of 99.42510082757477%.

7.DISCUSSION AND RECOMMENDATION FOR FUTURE DEVELOPMENT

Same model and techniques can be used for various other organizations like Netflix, Samsung etc to analyze there their stock prices and predict them for the future.

so let's now see what is the future scope for this project....

- **Feature Engineering:** To improve predictive accuracy, consider incorporating additional features such as trading volume, technical indicators, and external factors that may influence stock prices.
- **Hyperparameter Tuning:** Experiment with different hyperparameters for the regression models. Techniques like cross-validation and grid search can help identify the best hyperparameters for each model.
- **Time Series Analysis:** Given that stock prices are inherently time series data, consider using time series forecasting methods like ARIMA or LSTM to capture temporal patterns.
- **Ensemble Models:** Explore ensemble methods such as Random Forest, Gradient Boosting, or stacking multiple regression models, as they can often yield better results.
- **Cross-Validation:** Instead of a simple train-test split, implement k-fold cross-validation to assess model stability and generalization performance more robustly.
- **Data Sources:** Expand data sources beyond a single stock. Building a portfolio prediction model or analyzing interdependencies between different stocks and asset classes can be valuable.
- **Risk Management:** Incorporate risk management strategies into the analysis, such as assessing risk-adjusted returns, setting stop-loss levels, and portfolio optimization.
- **Real-Time Data:** Modify the code to work with real-time data from financial APIs, enabling continuous prediction and trading strategies. Consider using libraries like Alpaca or Interactive Brokers API for real-time data access.
- **Documentation and Comments:** Improve code documentation and add comments to explain complex sections, making the code more accessible and easier to maintain.
- **Deployment:** If the goal is to use the model for trading, consider deploying it as a web application or using algorithmic trading platforms.
- **Ethical Considerations:** Be aware of the ethical implications in financial modeling, and ensure that models are not used for unethical purposes or market manipulation.

- **Regular Updates:** Stock market conditions change over time. Regularly update the model with new data and reevaluate its performance to ensure it remains relevant and accurate.

Overall, this code provides a good starting point for stock price prediction, but there is ample room for improvement and customization, especially for practical applications in financial markets.

8.CONCLUSION

This project successfully implemented and evaluated three regression models for stock price prediction, with Linear Regression being the top performer. The models demonstrated a high degree of accuracy, achieving an R-squared score of around 99.43%. The project offers a solid foundation for further development, including feature engineering, model tuning, and consideration of ethical implications. It provides valuable insights for those interested in stock price prediction and analysis in the financial domain.