Apple Stock Price Prediction: Evaluation of Multi - Model Performance

Members:

24109263g LI Zhecheng

24131632g LI Yuyang

24103072g HUANG Yue

24057076g PAN Yutong

1. Introduction

In the complex and ever-changing financial market, stock price prediction has long been an in-depth research topic with extremely high practical value. The closing price of a stock is a key indicator that reflects the overall market sentiment at the end of each trading day and the comprehensive judgment of investors.

This project aims to use three distinct and widely applicable methods to predict stock closing prices: decision trees, long short-term memory networks (LSTM), and autoregressive integrated moving average model (ARIMA). Decision trees have a simple and easy to understand tree structure that can handle nonlinear relationships in data and provide clear decision rules. LSTM, as a special type of recurrent neural network, aims to capture long-term dependencies in sequential data, making it particularly suitable for analyzing time series data such as stock prices. ARIMA, as a traditional time series forecasting model, can effectively model the autocorrelation and seasonality in historical stock price data.

By comparing the prediction results of these three methods, this project has multiple important objectives. Firstly, it attempts to find the most accurate and reliable stock closing price prediction model among these three methods, which market participants can directly apply to improve the accuracy of investment decisions. Secondly, through the comparison process, we can gain a deeper understanding of the advantages and disadvantages of each method in processing stock price data. These studies can provide guidance for future research on stock price prediction, such as inspiring the development of hybrid models that combine the advantages of different methods. In summary, this project not only contributes to the practical field of stock investment, but also enriches the academic research on time series prediction methods in the context of financial data.

2. Data Source and Transformation

The data source for this analysis is a dataset from Kaggle, which provides daily historical data of Apple Inc. stock (stock code: AAPL) in US dollars. Apple Inc. is a well-known American multinational technology company that has been recognized as the world's highest revenue technology company and the most valuable company since January 2021.

The key data involved in the prediction is the numerical columns in the Apple stock dataset, as these columns are likely to contain key information such as stock prices and trading volumes, which can have a significant impact on the prediction results.

In order to obtain useful data, several ETL (Extract, Transform, Load) processes are executed. Firstly, in the data cleaning step, the missing values are delected. This method fills missing values with the last known value in the column, ensuring data continuity. Next, in order to address potential outliers that may distort the analysis results, the Z-score method is applied. Use the custom function remove_outliers_zscore to filter out data points in numerical columns with absolute Z-scores exceeding the threshold of 3. This effectively removes extreme values from the dataset. Finally, all numerical columns are normalized using MinMaxScaler. Normalization scales data to a fixed range (usually between 0 and 1), which can improve the performance of machine learning algorithms and make different features more comparable. Through these processes, the dataset is transformed into a format that is more suitable for subsequent analysis and prediction.

3. Analysis Steps and Trials

3.1 Decision Tree

3.1.1 Model Definition

A Decision Tree is a non-linear predictive modeling tool that, through a set of binary rules, splits a dataset into branches to form a tree of decisions. This tree structure consists of nodes that form the decision points and leaves that represent the outcomes. Our decision tree model applies these concepts to predict AAPL's stock closing price based on daily trading data like opening price, high, low, and volume.

3.1.2 Model Building

The construction of our model began with loading and preprocessing the data from AAPL_preprocessed.csv, which includes ensuring that the 'Date' column was in the proper datetime format and sorting the data chronologically. This preprocessing step is crucial as it assures that the sequence of data points is maintained, which is particularly important for time-series data, although not directly utilized in decision tree learning.

The features selected for predicting the closing price are 'Open', 'High', 'Low', and 'Volume'. These features are commonly used in stock price prediction due to their direct impact on the stock's closing price. The dataset was split into training (80%) and testing (20%) sets, ensuring a robust set for training while retaining an untouched subset for model validation.

3.1.3 Training Process

The Decision Tree Regressor's training process involves learning from historical data to predict future outcomes, which, in our context, is the AAPL stock closing price. Below outlines the specific steps taken in training our model:

1. Data Preparation: Initially, the AAPL stock dataset was loaded and preprocessed. This included converting the 'Date' column to a datetime format and sorting the data chronologically to ensure the integrity and usability of the time series data, even though time series order is not directly utilized by decision trees.
2. Feature Selection: The features selected for the model—'Open', 'High', 'Low', and 'Volume'—are typical predictors used in stock price forecasting due to their direct influence on the closing price.
3. Splitting Data: The dataset was divided into training (80%) and testing (20%) sets using the train_test_split function. This separation is crucial to validate the model's performance on unseen data, ensuring an unbiased evaluation.
4. Model Initialization and Training: A Decision Tree Regressor was initialized with a specified random state to ensure reproducibility. The model was then trained on the training set, where it learned to predict the closing price based on the patterns observed in the feature set.

During training, the decision tree algorithm recursively splits the training data by selecting the best split at each node based on the MSE criterion. It continues splitting until it fully captures the relationships in the data or until further splitting does not significantly reduce the error, thus forming a tree structure.

3.1.4 Prediction Process

Once the model was trained, the prediction phase involved:

1. Applying the Model: The trained Decision Tree Regressor was used to predict the closing prices on the unseen test data. This step tests the model's generalization capability and its effectiveness in capturing and utilizing the learned patterns.
2. Generating Predictions: For each instance in the test set, the model followed the decision paths established during training, from the root to a leaf node, where each path's decision is based on the input features' values. The value at the leaf node gives the predicted closing price for that particular day.

3.2 LSTM

3.2.1 Model Definition

Long Short-Term Memory (LSTM) is a special type of recurrent neural network (RNN), which was proposed by Hochreiter and Schmidhuber in 1997 to solve the problem of gradient disappearance or gradient explosion when traditional RNN processes long sequence data. LSTM can selectively retain or discard information by introducing gating mechanisms (input gates, forget gates, output gates) and Cell states, thus effectively capturing long-term dependencies in time series.

The core structure of LSTM includes:

1. Forget Gate: Determines what information in the cell state needs to be discarded.
2. Input Gate: Updates the cell status and filters the current input of valid information.
3. Output Gate: Generates the output of the current time step based on the cell state and the current input.

Because of its forecast for time series data with strong performance, we choose LSTM to forecast the stock price

3.2.2 Model Building

We construct a Two-Layer LSTM prediction model with a historical time step of 60 days and feature selection as opening price, high price, low price and volume, then predict close price.

In the first layer LSTM, we have 128 neurons, and in the second layer LSTM, we have 64 neurons. At the same time, a dropout rate of 0.3 is added after both layers of LSTM to prevent overfitting.

3.3 ARIMA

3.3.1 ARIMA(Autoregressive Integrated Moving Average Model)

(1) Parameters and mathematical forms of ARIMA

ARIMA model is a statistical model for time series analysis and forecasting, which combines three methods: autoregressive (AR), difference (I) and moving average (MA), and is suitable for processing time series data with trend component or seasonal component.

The ARIMA model has three parameters:p,d,q.

- p- the lags of the time series itself in the forecasting model, also known as the AR/Auto-Regressive term
- d- the time series data needs to be differentiated by several orders of difference to be stable, also known as Integrated item.
- q- lags of the prediction error in the prediction model, also known as the MA/Moving Average term.

The basic principle of the ARIMA model can be expressed by the following formula: ARIMA(p, d, q) = AR(p) + I(d) + MA(q).

Suppose we know (p, q, d), ARIMA is mathematically expressed as:

$$\widehat{y_t} = \mu + \emptyset_1 \times y_{t-1} + \ldots + \emptyset_p \times y_{t-p} + \theta_1 \times e_{t-1} + \ldots + \theta_q \times e_{t-q},$$

where Φ represents the coefficient of AR and θ represents the coefficient of MA.

3.3.2 SARIMAX

SARIMAX model combines the effects of seasonal autoregressive moving average models (SARIMA) and exogenous variables (X) to more accurately predict and simulate time series data. The advantage of the SARIMAX model is that it can process time series data with seasonality and trend, and it can consider the influence of external factors on the time series.

SARIMAX model has strong modeling ability and forecasting accuracy, so it is suitable for predicting stock data.

3.3.3 Modeling and Training

(1) Dividing Training and Test Set

Training_size is set to be -30, which means the close of last 30 days is the test set, and data of the remaining days is set to be the training set.

(2) Parameters Selection

We used the method of 'auto_arima' to automatically select the optimal ARIMA parameters. Seasonal is set to be false, as the dataset is considered to be seasonally unrelated.

After the training process, according to the search procedure in Picture 1, we found the best parameters to be (5, 2, 0), which means the lag is 5 days, the difference degree is 2 and moving average wasn't considered.

```
Performing stepwise search to minimize aic
 ARIMA(2,2,2)(0,0,0)[0] intercept   : AIC=inf, Time=9.74 sec
 ARIMA(0,2,0)(0,0,0)[0] intercept   : AIC=-1971.687, Time=0.94 sec
 ARIMA(1,2,0)(0,0,0)[0] intercept   : AIC=-4411.342, Time=1.02 sec
 ARIMA(0,2,1)(0,0,0)[0] intercept   : AIC=inf, Time=5.16 sec
 ARIMA(0,2,0)(0,0,0)[0]             : AIC=-1973.687, Time=0.25 sec
 ARIMA(2,2,0)(0,0,0)[0] intercept   : AIC=-5391.545, Time=1.23 sec
 ARIMA(3,2,0)(0,0,0)[0] intercept   : AIC=-6038.963, Time=2.48 sec
 ARIMA(4,2,0)(0,0,0)[0] intercept   : AIC=-6325.941, Time=2.16 sec
 ARIMA(5,2,0)(0,0,0)[0] intercept   : AIC=-6687.294, Time=2.79 sec
 ARIMA(5,2,1)(0,0,0)[0] intercept   : AIC=inf, Time=14.25 sec
 ARIMA(4,2,1)(0,0,0)[0] intercept   : AIC=inf, Time=10.89 sec
 ARIMA(5,2,0)(0,0,0)[0]             : AIC=-6689.292, Time=0.90 sec
 ARIMA(4,2,0)(0,0,0)[0]             : AIC=-6327.940, Time=0.99 sec
 ARIMA(5,2,1)(0,0,0)[0]             : AIC=inf, Time=6.21 sec
 ARIMA(4,2,1)(0,0,0)[0]             : AIC=inf, Time=3.99 sec

Best model:  ARIMA(5,2,0)(0,0,0)[0]
Total fit time: 63.002 seconds

最优模型参数: (5, 2, 0)
```

Figure 1

(3) Established model

The summary of the ARIMA model is shown in Picture 2 below:

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                 8899
Model:               SARIMAX(5, 2, 0)   Log Likelihood                3350.646
Date:                Mon, 31 Mar 2025   AIC                          -6689.292
Time:                        11:58:21   BIC                          -6646.731
Sample:                             0   HQIC                         -6674.802
                               - 8899
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.8174      0.004   -228.154      0.000      -0.824      -0.810
ar.L2         -0.6634      0.004   -149.535      0.000      -0.672      -0.655
ar.L3         -0.5135      0.005    -98.365      0.000      -0.524      -0.503
ar.L4         -0.3352      0.005    -67.223      0.000      -0.345      -0.325
ar.L5         -0.2003      0.004    -50.305      0.000      -0.208      -0.192
sigma2         0.0276      0.000    266.968      0.000       0.027       0.028
===================================================================================
Ljung-Box (L1) (Q):                   8.32   Jarque-Bera (JB):            435166.97
Prob(Q):                              0.00   Prob(JB):                         0.00
Heteroskedasticity (H):            1056.65   Skew:                            -0.13
Prob(H) (two-sided):                  0.00   Kurtosis:                        37.26
===================================================================================
```

Figure 2

- P>|z|: P-value that is used to determine whether the parameter estimates are statistically significant. Every P-value is 0.000 and less than 0.05, which is considered significant.
- [0.025, 0.975]: Upper and lower limits of the 95% confidence interval, giving the range of parameter estimates.
- sigma2: The square of the standard deviation of the residual. The model has a relatively small square of the standard deviation of the residual, which means that there is little difference between the predicted value of the model and the observed value, so the model has a relatively good fit.

4. Model Prediction and Evaluation

4.1 Decision Tree

4.1.1 Model Prediction

The visualization, as shown in the plot "Actual vs Predicted Close Prices on Test Set", provides a visual representation that mirrors the statistical findings:

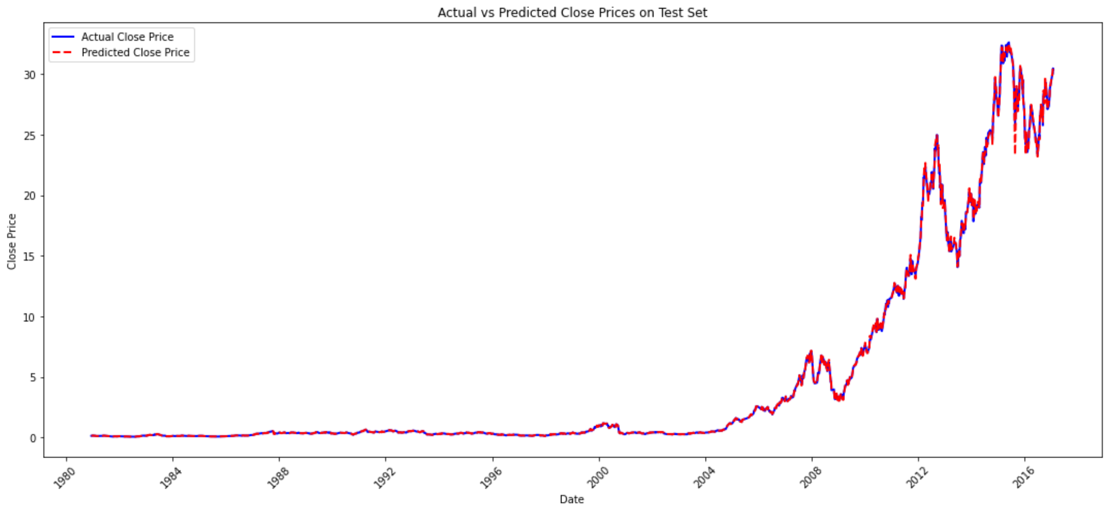

Figure 3 Actual vs Predicted Close Prices on Test Set

1. Graph Analysis: Both the actual (blue line) and the predicted (red dashed line) close prices are plotted against the dates. The lines are closely overlaid for most parts of the graph, which is a clear visual indicator of the model's high accuracy.

2. Trend Observation: From the start of the time series up to the most recent date, both lines follow a similar trajectory with matching peaks and troughs. The closeness of these lines through different market phases—ranging from steady states to volatile periods—illustrates the model's robustness.

3. Critical Insight: The close alignment throughout the time series, particularly during rapid rises and falls, suggests that the model is capable not only of tracking the general trend but also of adapting to sudden changes in the market conditions.

4.1.2 Model Evaluation

Evaluation Metrics:

After training the Decision Tree Regressor, we have rigorously evaluated its performance using several statistical metrics to ensure a comprehensive assessment:

Table 1

Metric	**Value**
MSE	0.0121
MAE	0.0394
MAPE	1.1531%

1. Mean Squared Error (MSE): The MSE of the model is 0.0121, indicating that on average, the squared difference between the actual and predicted stock prices is very low. This small value suggests that the model predictions are very close to the actual values, showing high accuracy.

2. Mean Absolute Error (MAE): The MAE is calculated to be 0.0394. This value represents the average magnitude of the errors between the predicted and actual values without considering their direction. The low MAE further corroborates the model's ability to predict the closing stock prices with minimal deviation from the true values.

3. Mean Absolute Percentage Error (MAPE): With a MAPE of 1.1531%, the model demonstrates excellent predictive reliability. This percentage quantifies the average absolute percentage difference between the predicted and actual prices, underscoring the model's efficacy in a more relatable percentage term, which is particularly useful for stakeholders.

4.2 LSTM

4.2.1 Model Prediction

We use the LSTM model built just now for training on the training set, and the training process is as follows:

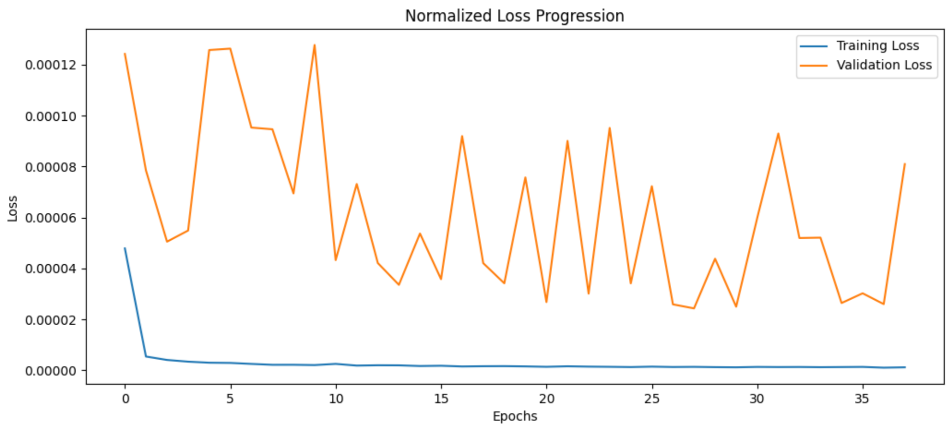

Figure 4

As can be seen from the picture above:

Both Training Loss and Validation Loss decrease with the increase of training rounds (Epochs), indicating that the model is learning and optimizing.

The two curves are very close, and both converge to a lower value (close to 0), indicating that the model has no obvious overfitting or underfitting problems.

The loss eventually leveled off, and the decrease was small in the late training period (after about 25 epoch), indicating that the model may have converged and the improvement from continued training is limited.

The overall performance is good, and the training and validation losses both reach a low level (about 0.00002), indicating that the model performs well on both training and validation data.

And we get the prediction:

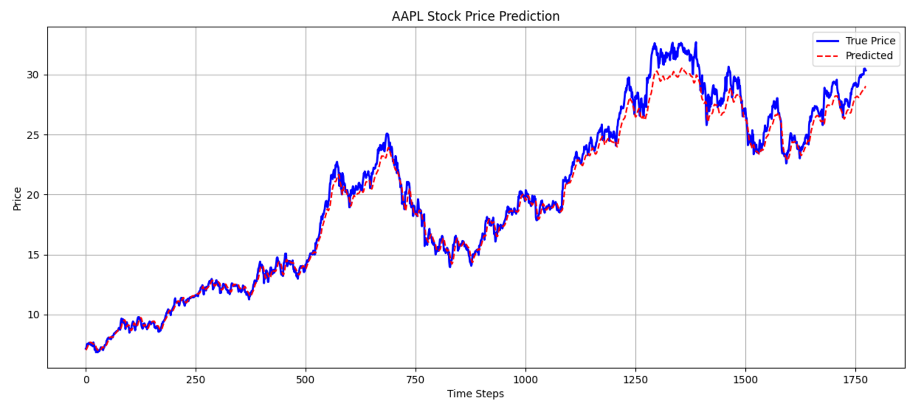

Figure 5

As can be seen from the picture above:

The trend of true price and predicted price is highly consistent, indicating that the model has good forecasting effect and can more accurately track the actual stock price changes.

The prediction curve is close to the real curve, especially in most time steps, where the two almost overlap, indicating that the model has a strong ability to fit historical data.

The forecast model has performed well on AAPL stock price and may be suitable for short - or medium-term trend analysis.

4.2.2 Model Evaluation

The performance metrics are shown in Table 2:

Table 2

Metric	**Value**
MSE	0.7681
RMSE	0.8764
MAE	0.6346

From the table, we can see that MSE, RMSE and MAE are all less than 1, indicating that the prediction error is acceptable. However, compared with ARIMA model and Decision Tree model, the prediction error is still larger, which may be because there are irregular fluctuations in the middle and late prediction. After the time step of 1250 in the prediction graph, there is a certain error between the prediction curve and the true value curve.

4.3 ARIMA

4.3.1 Model Prediction

(1) Rolling Forecasting

We used the method of rolling forecasting to make step-by-step forecasts of closing prices over the next 30 days. The core idea of rolling forecasting is to use current and previous data to predict the future, and to update historical data and models as new observations become available. The process is as follows:

- Initial state: The historical data ‘history’ contains all the data for the training set.The test set is the future data to be predicted, where the true value at each point in time is unknown at the time of prediction, but in code implementation, the true value is usually assumed to be known for model update.

- Step 1 Prediction: Using the history data in training model, call `model.predict(n_periods=1)` to predict a value for the 1st time point in the future. Then, save the predicted value and add the true value from the first point in time in the test set to history and update the model with true values.
- Step 2 Prediction: Update the model using the updated history, and predict the value of the 2nd time point in the future, and the remaining process is the same as Step 1.
- Repeat the above steps until all time points have been predicted.

(2) Prediction

The result of prediction is shown in Picture 3 below:

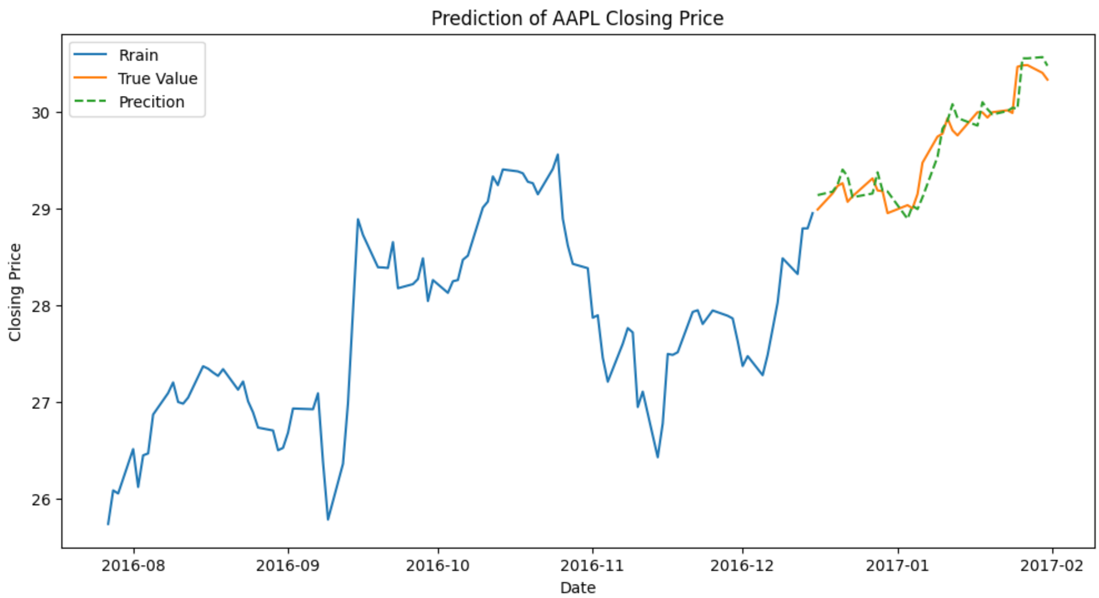

Figure 6

After the beginning of 2017, the predicted value and the true value basically coincide, indicating that the model has a good forecasting performance in this period of time, and can accurately capture the trend of stock prices.

However, at the end of 2016, the gap between the predicted value and the true value was more pronounced, indicating the predictive performance of the model in the early stages still have room for improvement.

4.3.2 Model Evaluation

The performance metrics are shown in Table 3:

Table 3

Metric	**Value**
MSE	0.0278
RMSE	0.1667
MAE	0.1295

RMSE and MAE both reflect the degree of deviation between the predicted value and the actual value. MAE shows that most of the prediction errors are small, but the value of RMSE reflects the sensitivity of the model to extreme fluctuations.

4.4 Comparison

In this study, Mean Squared Error (MSE) was selected as the primary evaluation metric to assess the performance of different models in predicting stock closing prices. The MSE values obtained for the three models are as follows: 0.7681 for the LSTM model, 0.0278 for the ARIMA model, and 0.0121 for the Decision Tree model.

A lower MSE value indicates a better fit of the model to the actual data, as it measures the average of the squares of the errors. As can be clearly seen from the values, 0.7681 > 0.0278 > 0.0121. This implies that the Decision tree model demonstrates the best predictive performance in forecasting stock closing prices among the three models considered. It is followed by the ARIMA model, and the LTSM model shows the relatively poorest performance.

Table 4

Metric	**Decision Tree**	**LTSM**	**ARIMA**
MSE	0.0121	0.7681	0.0278
RMSE	0.0394	0.8764	0.1667
MAE	1.1531%	0.6346	0.1295

5. Conclusion

This report compared Decision Tree, ARIMA, and LSTM models for predicting stock closing prices. In conclusion, based on the MSE evaluation, the Decision tree model is the most suitable for predicting stock closing prices in the prediction of stock price, while the ARIMA model also provides a decent performance.

However, the LSTM model may require further optimization or alternative model selection for more accurate stock price predictions. As can be seen from the prediction results, there may be slight lag or deviation in the prediction of AAPL stock closing price by LSTM model. In some local areas (such as those with severe fluctuations), the predicted value may be slightly delayed or slightly deviated, but the overall trend is consistent.

Recommendations for Future Work:

1. Decision Tree Model

Parameter Tuning: Adjust tree depth and minimum samples per leaf to improve accuracy and avoid overfitting.

Feature Engineering: Incorporate technical indicators (e.g., moving averages, RSI) and macroeconomic factors to boost predictive power.

2. LSTM Model

Optimization: Use grid search for optimal parameters or switch to BiLSTM to reduce lag and deviation in AAPL price prediction, especially during high - volatility periods.

3. ARIMA Model

Parameter Refinement: Analyze different orders of differencing, autoregressive, and moving average terms for better data fit.

Missing Values: Apply interpolation instead of deletion when handling missing data.

4. Ensemble Techniques

Implement Random Forests or Boosted Trees to enhance prediction stability and accuracy across all models.