Proceedings of the 2nd International Conference on Financial Technology and Business Analysis
DOI: 10.54254/2754-1169/56/20231103

Stock Price Prediction for Technology Company

Yuchen Wang[1,a,*]

[1]*Graduate School of Arts and Sciences, Columbia University, New York, US*
a. yw3890@columbia.edu
**corresponding author*

Abstract: Individuals aim to develop accurate models for stock prices to make informed decisions as investors, so they can determine opportune moments to purchase and sell stocks for maximizing profits. This paper select Apple stock from yahoo finance range from Aug 1st 2013 to Aug 1st 2023, and then forecasting it's future 30 days stock price. This study contain four models, which are XGboost, linear regression, K-Nearest Neighbors (KNN) and Long Short-Term Memory (LSTM). Those models are all fit the train and test data and then draw a visualization plot. For selecting the best model, this paper use root mean squared error(RMSE) metrics and mean absolute percentage error(MAPE) and got the best model are Linear Regression and LSTM. Depending on the mechanism of four models, LSTM can treated as the best one to predict future stock price. In future study, researchers can use LSTM model more to predict stock price for other companies in order to get the best result.

Keywords: LSTM, prediction, RMSE, visualization

1. Introduction

Stock market prediction is important. It offers a range of advantages for investors and businesses alike. These benefits include the potential for profitable trading decisions, effective risk management by anticipating market downturns, informed strategic planning, and optimized portfolio construction. Accurate predictions also enable a competitive edge, drive research and development of innovative strategies, provide insights into market sentiment, and inform long-term financial goals.

Technology company's stock price is sensitive to many factors like constant innovation, uncertain profit prospects, susceptibility to market expectations and sentiment and so on. So how to predict their stock price is the essential part in investment activity.

In the past, many researchers using single machine learning method like decision tree [1-3], linear regression [4-7], support machine vector [7-10] and so on to predict the stock price. There prediction is easy, but it might not perform very accurately for prediction part. Thus, this paper uses XGboost, KNN, linear regression and LSTM to analysis the issue so as to fill the potential research gap.

2. Data

The data collected for this research is from Yahoo finance. The Stock data is "AAPL". The reason for the selection is shown below. Apple stands as a renowned global technology powerhouse, celebrated for its groundbreaking hardware and software offerings. Among its array of products are the iconic iPhone, iPad, Mac, and Apple Watch, complemented by an array of software and services. Apple also conducts research and development in fields such as artificial intelligence, augmented

DOI: 10.54254/2754-1169/56/20231103

reality, and autonomous driving. The data range is from August 1st, 2013, to August 1st, 2023, which is a ten-year range. The reason for this range is that using long range data can reduce the percentage of outlier, making the price prediction more accurately. For the summary chart, close price is the only value the study need. The mean closing price is 71.00. The min price is 16.08, and the max price is 196.45, standard deviation is 52.92, 25%, 50%, 75% quartile are 28.48, 44.52 and 125.30 (See Table 1).

Table 1: Summary Statistics for AAPL Closing Price.

	Summary Statistics	Close
1	count	2516.0000
2	mean	71.003268
3	std	52.920136
4	min	16.075714
5	25%	28.479375
6	50%	44.517500
7	75%	125.297499
8	max	196.449997

From the data visualization plot, the information suggests that the min price is in 2014 and the max price is in 2023. This infer that AAPL is developed dramatically in the past ten year as the closing price is rising (See Figure 1).

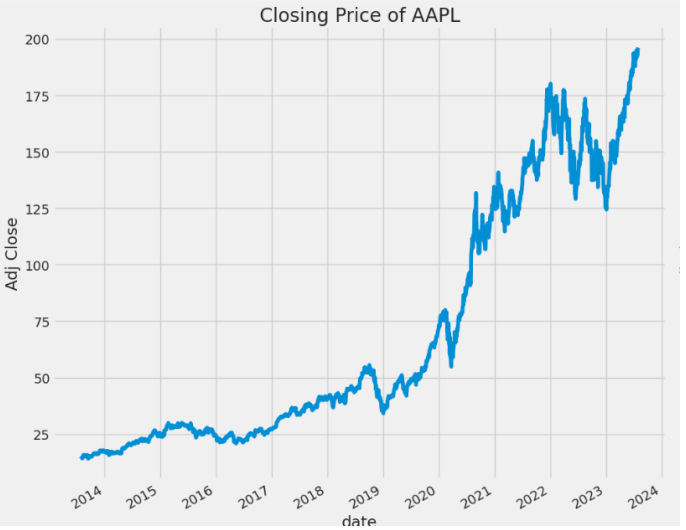

Figure 1: Closing Price for AAPL.

3. Method

3.1. XGboost

XGBoost is based on the concept of gradient boosting, which involves sequentially training a series of weak learners like decision trees and combining their predictions to improve the overall model's accuracy. XGBoost employs an efficient gradient optimization algorithm to minimize the loss function during the training process. This optimization approach accelerates convergence and enhances training efficiency.

DOI: 10.54254/2754-1169/56/20231103

XGBoost is often used for predicting stock prices due to its ability to handle complex relationships, capture non-linearity, and effectively incorporate various features that influence stock price movements. Its ensemble learning approach, regularization techniques, and handling of missing data make it suitable for the dynamic and noisy nature of financial data.

The steps for XGBoost for stock price prediction:

Data Splitting: Divide the dataset into three distinct sets: training, validation, and testing.

Model Selection and Training: use hyperparameters to tune the model in order to optimize model performance. Common parameters include learning rate, maximum depth, number of boosting rounds, and regularization terms. Utilize the training data to train the XGBoost model, followed by validation using the validation dataset to assess its performance.

Hyperparameter Tuning: hyperparmaters have many combinations. In order to find the best one, the study need to use some functions like grid search or random search.

Evaluation: Use some metrics like RMSE. Those can assess whether the model is fitted.

3.2. KNN

The KNN algorithm is a straightforward and intuitive machine learning technique employed for tasks such as classification and regression, even extending to predicting stock prices. KNN's predictions hinge on how closely input data points resemble established labeled data points present in a training dataset. KNN is a relatively simple algorithm to understand and implement. It doesn't involve complex mathematical concepts or intricate hyperparameter tuning. And it has the ability to determine whether the data relationships are linear. Additionally, KNN have the ability potentially adapt to changing market conditions as it considers recent similar data points for predictions.

Here are steps in predicting the stock price:

Prepare the dataset with features and the target variable. Then make the dataset separated to train set and test set.

Choose one of the K values that can represents the neighbors' number that will be considered when making predictions. This value is typically chosen through cross-validation.

Identify one of the training points, which has the smallest range to the testing point.

For regression purposes, compute the mean of the target values belonging to the KNN. In the case of classification tasks, identify the most frequent class within the KNN.

By applying some metric like RMSE, the study can assess the model performance.

3.3. Linear Regression

Linear regression is the tool to many things like predicting the stock price because of its simplicity, ability to identify trends, and interpretability. It serves as a baseline model and provides insights into the relationships between predictor variables and stock prices.

Using linear regression for predicting stock prices is another approach, though it comes with its own set of considerations and challenges. Linear regression assumes a linear relationship between the input features and the target variable, which in this case would be the stock price. Here's how it might approach using linear regression for stock price prediction:

Here are steps in predicting the stock price:

Separating the Data: Dataset is separated to training, testing and validation sets to train the study, tune hyperparameters, and evaluate performance.

Model Selection and Training: choose linear regression as the modeling technique. Train the model using the training data. This technique can lower the error for the predicted VS actual values.

DOI: 10.54254/2754-1169/56/20231103

Feature Selection: given that linear regression assumes a linear relationship, it's important to select features that are more likely to exhibit linear correlations with the target. Feature selection techniques can help in choosing the most relevant features.

Evaluation: assess the model's performance using appropriate evaluation metrics like MAE, MSE, RMSE and R-squared.

Prediction: apply the trained linear regression in order to have a accurate predicted result in the future data.

3.4. LSTM

LSTMs are effective for dealing with sequential data like stock prices because it can capture long-range dependencies, allowing them to consider historical data's influence on future prices. Also, it can identify and adapt to both short-term fluctuations and long-term trends. Lastly, it trained on a large dataset can capture general financial market patterns for specific prediction tasks.

The LSTM cell has several components in the whole working phase:

1. The input gate (i_t) assesses the importance of incoming data in relation to the existing cell state.
2. The forget gate (f_t) decides which information in the previous cell state should be discarded or ignored.
3. During the cell state update (g_t) phase, new potential values for the cell state are calculated based on the input.
4. The cell state (C_t) is then modified using the combined influence of the input gate, forget gate, and cell state update.
5. The output gate (o_t) determines how much the current cell state contributes to the final output of the LSTM.
6. The hidden state (h_t) is computed using the output gate and the updated cell state, resulting in the new representation of the LSTM cell (Details are shown in Figure 2).

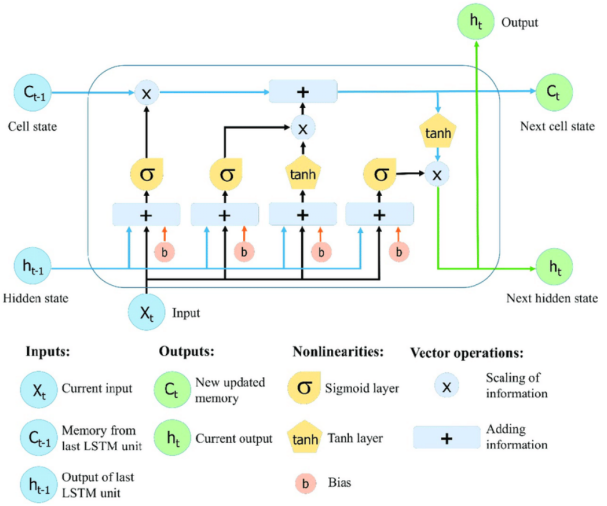

Figure 2: LSTM Model.

4. Result

4.1. XGboost

In XGboost model, ['High', 'Low', 'Open', 'Volume'] is the input variable, and the [‘Close’] is the target variable. Then the study split the data to 80 percent of training data and 20 percent of testing

DOI: 10.54254/2754-1169/56/20231103

data. Afterwards, the training process involves utilizing the XGBRegressor function. This involves fitting the model using the training data, allowing the algorithm to learn from the provided information. Subsequently, the trained model is used to make predictions on the test data, enabling it to generalize its learned patterns to new, unseen data. The prediction is stored as y_pred. To check the model accuracy, the study uses rmse. And the rmse is 39.30 (See Figure 3).

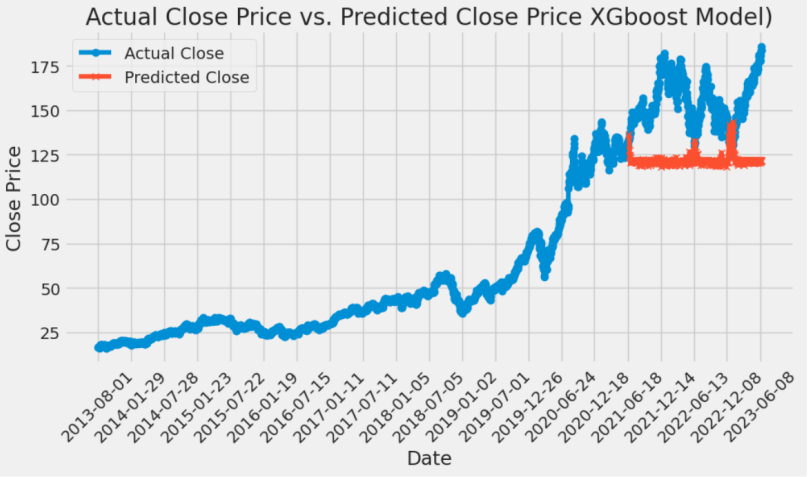

Figure 3: Actual Close Price VS Predicted Close Price in XGboost Model.

4.2. KNN

The dataset is separated from the feature variables ('High', 'Low', 'Open', 'Volume') into X and the target variable ('Close') into y. Then the study defined a TimeSeriesSplit object to perform time series cross-validation with two splits, using the StandardScaler to standardize the feature variables in both the training and testing sets. By performing hyperparameter tuning using cross-validation with the number of neighbors (K) as the hyperparameter, the study plotted the error (mean squared error) for different values of K to help us find the best K value. After that, GridSearchCV can be used, and the best K value is 46. Then the best_knn model is using this value, and then the study fit the best_knn model on the scaled training data. Then the best_knn is used to predict our scaled data. The best_knn model plotted a graph representing actual values VS predicted value for test data. After that, the RMSE is calculated to be 23.25 (See Figure 4).

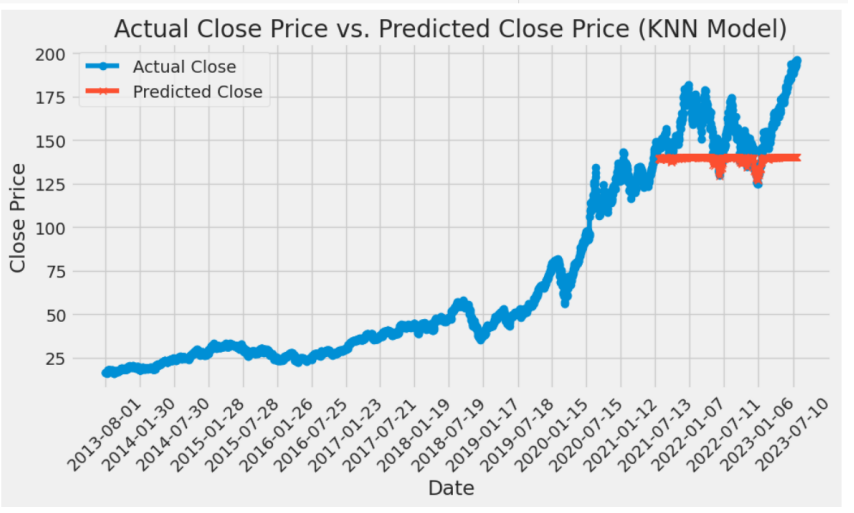

Figure 4: Actual Close Price VS Predicted Close Price for KNN Model.

DOI: 10.54254/2754-1169/56/20231103

4.3. Linear Regression

For data preprocessing, the future price is calculated using historical price data and stored it in the data, then it dropped the last 30 rows as the data don't have future data to validate these predictions. In linear regression, the study makes ['High', 'Low', 'Open', 'Volume'] to input variable and ['Predicted'] as target variable. Then the data is split to train data and test data then standardized it. After that, in order to train the model, the model used the function LinearRegression() to instantiate the model, and it trained the model using scaled data. For prediction, the "model.predict()" function takes the scaled test set features 'X_test_scaled' as input and returns the predicted target variable values (in this case, future prices) based on the trained model's learned patterns. The graph for predicted value vs actual value is plotted. The root mean squared error for 15.21 (See Figure 5).

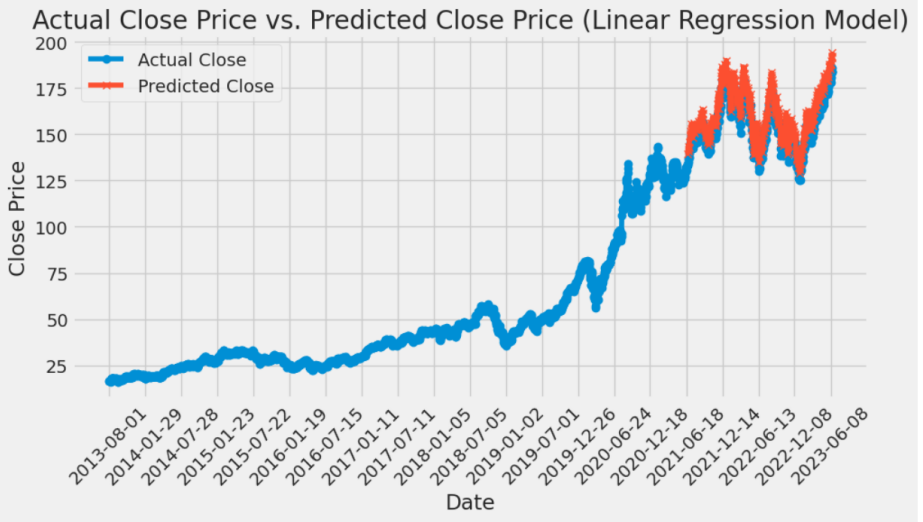

Figure 5: Actual Close Price VS Predicted Close Price for Linear Regression Model.

4.4. LSTM

LSTM has many steps within the model. The first step is same as previous mode for splitting the data, then use function to convert data to LSTM format. Generally, the model use time steps equal to 100. Next, the model is defined with three layers and one dense layer. The optimizer was set as 'Adam', the loss function was set as 'mean_squared_error', and the evaluation metric was set as Mean Absolute Percentage Error (MAPE). In this training, the epochs equal to 10 and the batch size equal to 32 and the training progress is displayed (verbose=1). The val_mape value for the last epoch is 5.9502, which is an acceptable value. After that, the model is iterated over the parameters to find the best model.

Next, the best model can use to predict the X_train and X_test by using predict function and then transform back to the original form. The RMSE for y_train and y_test is 60.84 and 160.69. For AAPL stock prediction, a visualization plot is made. The train predict price and the test predict price has little difference from the actual price but not a lot (See Figure 6).

DOI: 10.54254/2754-1169/56/20231103

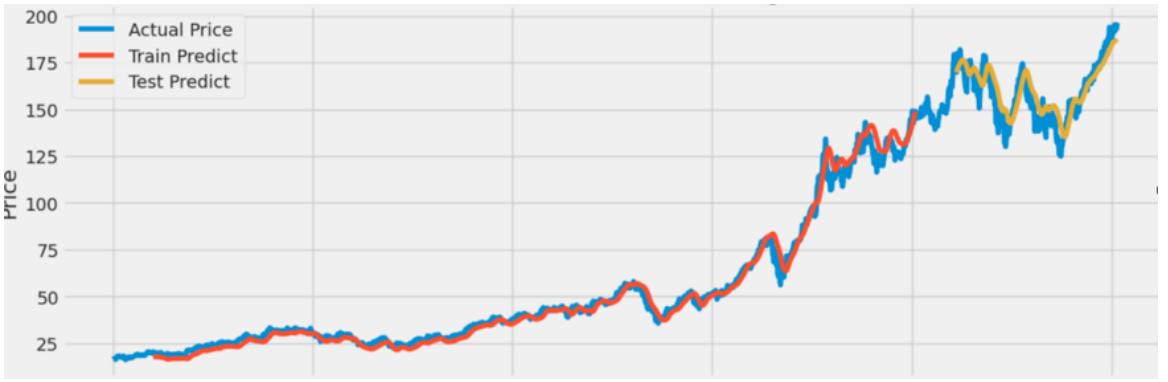

Figure 6: Actual Price VS Predicted Price for LSTM.

5. Conclusion

Among the models employed, both LSTM and linear regression exhibited superior performance in terms of accuracy. Linear regression showcased commendable predictive power, displaying a lower value for root mean squared error. This translated to a remarkable accuracy level in the outcomes. In contrast, the K-Nearest Neighbors (KNN) and XGboost model yielded a comparatively higher root mean squared error than Linear regression, thereby falling short in accuracy. XGboost, on the other hand, demonstrated the weakest performance, yielding a high root mean squared error of 15.21 and an underwhelming return.

This study face limitations including their struggle with non-linear and non-stationary stock data, the challenge of incorporating market sentiment and news, susceptibility to overfitting due to limited data, inability to grasp causality, and difficulty adapting to abrupt market shifts.

References

[1] *Wang, Y., and Guo, Y. (2020). Forecasting method of stock market volatility in time series data based on mixed model of ARIMA and XGBoost. China Communications, 17(3), 205-221.*

[2] *Vuong, P. H., Dat, T. T., Mai, T. K., and Uyen, P. H. (2022). Stock-price forecasting based on XGBoost and LSTM. Computer Systems Science & Engineering, 40(1).*

[3] *Han, Y., Kim, J., & Enke, D. (2023). A machine learning trading system for the stock market based on N-period Min-Max labeling using XGBoost. Expert Systems with Applications, 211, 118581.*

[4] *Stock Prediction Using Linear Regression. (2020) Retrieved from https://medium.com/analytics-vidhya/stock-prediction-using-linear-regression-cd1d8351f536*

[5] *Panwar, B., Dhuriya, G., Johri, P., Yadav, S. S., and Gaur, N. (2021). Stock market prediction using linear regression and SVM. In 2021 International Conference on Advance Computing and Innovative Technologies in Engineering, 629-631.*

[6] *Gharehchopogh, F. S., Bonab, T. H., and Khaze, S. R. (2013). A linear regression approach to prediction of stock market trading volume: a case study. International Journal of Managing Value and Supply Chains, 4(3), 25.*

[7] *Ravikumar, S., and Saraf, P. (2020). Prediction of stock prices using machine learning (regression, classification) Algorithms. In 2020 International Conference for Emerging Technology, 1-5.*

[8] *Kurani, A., Doshi, P., Vakharia, A., and Shah, M. (2023). A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting. Annals of Data Science, 10(1), 183-208.*

[9] *Sheth, D., and Shah, M. (2023). Predicting stock market using machine learning: best and accurate way to know future stock prices. International Journal of System Assurance Engineering and Management, 14(1), 1-18.*

[10] *Panwar, B., Dhuriya, G., Johri, P., Yadav, S. S., and Gaur, N. (2021). Stock market prediction using linear regression and SVM. In 2021 International Conference on Advance Computing and Innovative Technologies in Engineering, 629-631.*