# Predicting Next Day Stock Price by using Supervised Machine Learning Models on Technical Indicators

Man Yi Yeung
*College of Engineering*
*Drexel University)*
Philadelphia, PA, United States
my442@drexel.edu

***Abstract*—This project focuses on next-day stock price prediction by applying machine learning models on 27 technical indicators of the past 5 days. The stocks that were investigated are IBM, Apple (AAPL) and Johnson and Johnson (JNJ). Stock price and technical indicator data for the past 20 years are used to train the models after k-fold cross-validation split with blocking, normalization with standard scaler or min max scaler, as well as principle component analysis for dimensionality reduction. The machine learning models that were tested are linear regression, Bayesian ridge regression, lasso regression, decision tree regression, random forest regression, support vector regression, and long short term memory. From measuring the root mean square error (RMSE) of the prediction with the actual stock price for the IBM stock, the best models are the long short term memory model, linear regression and Bayesian ridge regression, with a RMSE score of around 2 to 5 for all price types of the IBM and JNJ stocks. The models work well for these two of the three stocks tested. For AAPL, this approach produces high RMSE values of more than 120, suggesting this stock's price trend does not have strong correlation with the exact past technical indicator combination chosen for this analysis. Thus, this technical indicator approach is not a generalized approach for all stocks, and may require adjustments for some stocks.**

## I. Introduction

The stock market is a significant part of the economy. Individuals, companies, and governments can influence and are impacted by changes in the stock market. Predicting stock price is a popular topic in machine learning. Stock investment strategies are very complex problems because it involves large amounts of data, and stock price movements are influenced by many factors from the the performance of the company, to the well-beings of related industries, and to the current situation of the government. Many research has been done in this area, and this project aims to predict the daily stock prices for three companies by using past technical indicators data gathered through Alpha Vantage API.

There are two main methods utilized in stock market prediction. The first one is fundamental analysis, where the stock price is compared to the actual worth of the company's stock. The estimated intrinsic value of the stock is based on the financial performance, company's business goals and so on. This approach is mainly used in long term stock prediction. The second approach is the technical analysis, where short term price trends are predicted from analyzing past pricing data and past technical indicator to evaluate price signals. Technical analysis is the approach used in this project, where past technical indicators data are used to predict the next day stock price.

## II. Related Work

There are many past research conducted in the area of stock price prediction. The approach and the machine learning models used differ as new prediction tools and concepts emerge.

One of the approach is artificial neural networks (ANN), a paper is published in 2004, using univariate neural networks approach to provide short term stock market predictions in order to evaluate profitability of different trading signals for S&P 500 stocks in the years from 1965 to 1999 [1]. Another study uses least squares support vector machine (LSSVM) method to predict stock market trend based on historical stock data [2]. This method evolves the input features through evolutionary algorithms or more specifically genetic algorithm. The genetic algorithm mimics Darwin's natural selection process in order to evolve inputted features to provide optimize prediction.

A term called "Stock2Vec" has been used to describe many algorithms that predicts stock price using different kinds of information of the stock market. One usage of this term is as a trained word embedding in a two-stream gated recurrent unit network approach to conduct sentiment analysis on financial news in order to stock price in the short run [3]. Another study uses "Stock2Vec" to represent the individual stock rather than word embedding. In this study, the embedding is trained as features and used with temporal convolutional network and deep learning in order to predict daily stock prices [4].

Past work that is similar to this project's approach includes a study of logistic regression using over 270 features of qualitative and quantitative information about the stock to predict the stock price trend as a classification problem of whether the stock is moving up, down or stationary [5]. There is also a similar paper, where artificial neural network (ANN), support vector machine (SVM), random forest and naive-Bayes are use on technical parameters calculated using stock

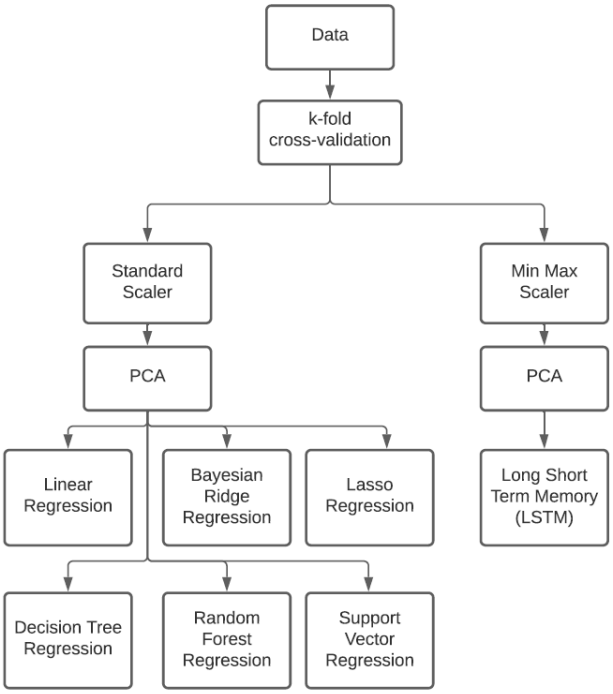



Fig. 1. Flow Chart of Machine Learning Process

trading data of open, high, low and close price, in order to predict stock price in the Indian market [6].

## III. Methodology

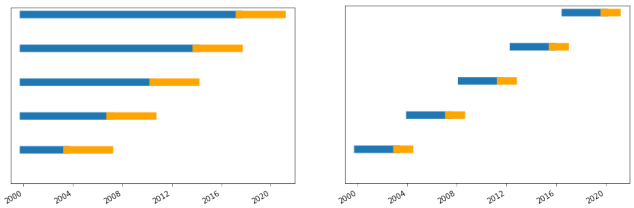
The project is completed using python 3.7.9 with Jupyter Notebook. It utilizes pandas, numpy, matplotlib libraries for data processing and plotting. Sklearn, Keras and TensorFlow are used for different implementing machine learning models.

Figure 1 shows a general flow chart of how the stock price and technical indicators data are pre-processed and fed to different machine learning models for price prediction.

### A. Data Collection

Stock pricing and technical indicators data are collected from Alpha Vantage API. The four types of daily stock price, open, close, high, and low, are the dependent variables of the analysis. The open and close price refers to the first and last bid price of the day, while the high and low price corresponds to the maximum and minimum price on a particular day.

The independent variables are the technical indicators over the past 5 days. There are a total of 27 technical indicators used including the volume of trade, dividend, simple moving average (SMA), exponential moving average (EMA), moving average convergence/divergence (MACD), stochastic oscillator (STOCH), relative strength index (RSI), average directional movement index (ADX), commodity channel index (CCI), Aroon indicator (AROON), Bollinger bands (BBANDS), Chaikin A/D line (AD), and on balance volume (OBV). (Note: Some of these indicators may contain multiple values all taken into account in the dependent variable dataset.)




(a) k Fold Without Blocking (b) k Fold With Blocking

Fig. 2. k Fold Time Series Split Without Blocking (on the left) and With Blocking (on the right), with k value = 5

These specific features are chosen because of their popularity and being marked with ”high usage” on Alpha Vantage API’s website. For some indicators, the number of data points used to calculate values such as moving averages need to be specified. Two values, 10 and 20 data points, are chosen to provide a better general picture for the next day stock price prediction.

### B. Data Pre-processing

There are three main steps in the pre-processing of the data, k-fold cross validation, data scaling, and principle component analysis (PCA).

*1) K-Fold Cross-Validation:* The first step is k-fold cross-validation. The reason of employing k-fold cross-validation instead of the normal train-test split is that the sample size with almost 20 years of data is very large, and it allows for models to be tested over different time range. Two types of k-fold cross-validation are used. One is the normal k-fold cross-validation for time-series data without shuffling and blocking. The second type has has blocking, which means only historical data up to some point is used to train the model, not all data prior to the train-test split. The difference between the two approaches can be visualized in Figure 2. In the pre-processing step of the data, a k values or the number of splits of 10 is used, but Figure 2, k value of 5 is used for easier visualization purpose.

*2) Data Scaling:* The second stage is scaling or standardization. As shown in Figure 1, the two types of scaling method used are standard scaler and min max scaler. Standard scaler enable the standardized value to have a mean of 0 and a standard deviation of 1. The min max scaler transform the data to have values between 0 and 1 only.

*3) Principle Component Analysis:* The last step before model implementation is Principle Component Analysis or PCA, aimed to reduce the dimensionality of the independent variable dataset. By performing PCA, the size of the dataset can be reduced, while preserving most of the information expressed by the original features. The extent of this reduction is determined by setting the amount of variance of the original features that the reduced dataset needs to explain. This threshold is set to 95%. The effects of PCA on the performance of various regression models used and the Long Short Term Memory model are studied, and the results are presented in the section IV. Experiments and Results IV.

| | linear_reg | bayesian_ridge_reg | lasso_reg | decision_tree_reg | random_forest_reg | support_vector_reg | LSTM |
|---|---|---|---|---|---|---|---|
| 0 | 4.808667 | 4.888521 | 11.106014 | 17.408132 | 17.483984 | 22.129125 | 13.876827 |
| 1 | 2.808404 | 2.801274 | 3.987724 | 7.786797 | 4.958296 | 4.456159 | 2.866125 |
| 2 | 3.101628 | 3.088688 | 2.980014 | 5.158098 | 3.653026 | 3.080102 | 2.010821 |
| 3 | 6.495706 | 6.500080 | 5.163131 | 10.160646 | 7.587659 | 10.179928 | 10.744795 |
| 4 | 13.141895 | 13.120353 | 3.175249 | 18.762084 | 18.259282 | 33.650716 | 7.194653 |
| 5 | 6.281452 | 6.241676 | 23.352842 | 29.532666 | 29.978575 | 56.429829 | 4.492716 |
| 6 | 5.765021 | 5.755637 | 15.123856 | 10.890031 | 7.693399 | 7.075891 | 4.440449 |
| 7 | 15.202643 | 15.195535 | 8.400180 | 11.456419 | 11.153010 | 11.324728 | 11.919901 |
| 8 | 21.318324 | 21.317838 | 15.665133 | 19.579172 | 17.881335 | 18.570012 | 20.087404 |
| 9 | 38.806086 | 38.774088 | 30.220179 | 36.295280 | 35.456723 | 31.534919 | 26.415850 |
| mean | 11.772983 | 11.768369 | 11.917432 | 16.702933 | 15.410529 | 19.843141 | 10.404954 |

(a) RMSEs of k Fold Without Blocking

| | linear_reg | bayesian_ridge_reg | lasso_reg | decision_tree_reg | random_forest_reg | support_vector_reg | LSTM |
|---|---|---|---|---|---|---|---|
| 0 | 3.032341 | 2.973207 | 2.889588 | 5.236390 | 3.195479 | 2.766315 | 3.809931 |
| 1 | 1.651575 | 1.644621 | 4.028208 | 8.876527 | 7.959286 | 2.921372 | 4.527879 |
| 2 | 1.020808 | 1.013065 | 1.447584 | 3.115549 | 1.609019 | 1.309962 | 1.811945 |
| 3 | 2.288440 | 2.287018 | 4.997377 | 6.849540 | 5.794320 | 5.467515 | 3.772171 |
| 4 | 2.706423 | 2.699042 | 2.476286 | 4.787787 | 3.422274 | 6.460802 | 3.946628 |
| 5 | 2.415856 | 2.420106 | 9.695940 | 17.550870 | 14.582419 | 17.573037 | 4.898388 |
| 6 | 2.199421 | 2.201165 | 3.459656 | 6.441903 | 4.099366 | 2.397456 | 3.688728 |
| 7 | 11.508884 | 11.461069 | 8.202754 | 10.266262 | 9.996224 | 9.716127 | 8.859591 |
| 8 | 2.176218 | 2.209733 | 10.536865 | 10.557376 | 10.590664 | 10.186672 | 7.553288 |
| 9 | 5.238736 | 5.255670 | 9.990838 | 11.820291 | 9.928072 | 6.025175 | 5.019330 |
| mean | 3.423870 | 3.416470 | 5.772510 | 8.550249 | 7.117712 | 6.482443 | 4.788788 |

(b) RMSEs of k Fold With Blocking

Fig. 3. RMSE values for predictions of various machine learning models on k Fold Time Series Split Without Blocking data (on the top) and on With Blocking data (on the bottom)

### C. Implementation of Supervised Machine Learning Models

Seven different supervised machine learning models are trained with pre-processed data to predict the next day stock price. They are linear regression, Bayesian ridge regression, lasso regression, decision tree regression, random forest regression, support vector regression, and long short term memory. They are all regression models except for long short term memory, which is a neural network model. The performances of each model are evaluated by calculating the root mean square error (RMSE) values between the actual historical stock price data and the model prediction.

## IV. Experiments and Results

The prediction on next day stock price was performed on three stocks for IBM, Apple ('AAPL') and Johnson and Johnson ('JNJ'). All four types of daily stock price, daily open, daily close, daily high, and daily low are forecasted for each company's stock.

For the results in the following sections, the price predictions and RMSE values are given for the IBM stock and for daily open price, unless explicitly written otherwise. The difference in predictions and RMSE values for the four price types are very small, and comparison among price types are shown in IV-C.

### A. Impact of k-fold Cross Validation with and without Blocking on Forecast Performance

For the "IBM" stock, the impact of whether to have k-fold cross-validation with or without blocking is studied with a k value of 10. The model prediction RMSE values for with and without blocking are presented in Figure 3.

| | linear_reg | bayesian_ridge_reg | lasso_reg | decision_tree_reg | random_forest_reg | support_vector_reg | LSTM |
|---|---|---|---|---|---|---|---|
| with PCA | 5.238736 | 5.255670 | 9.990838 | 11.667019 | 9.928072 | 6.025175 | 4.416079 |
| no PCA | 10.437316 | 4.605439 | 11.991744 | 9.074535 | 8.717152 | 6.015950 | 6.612826 |
| absolute difference | 5.198581 | 0.650231 | 2.000906 | 2.592484 | 1.210919 | 0.009225 | 2.196747 |

Fig. 4. Comparison of RMSE values for prediction using PCA and without using PCA before apply machine learning models

In Figure 3, k-fold cross validation with blocking results in mean RMSE values of between 3 and 10 for each model and a maximum RMSE of no more than 18. For result obtained without blocking k-fold cross validation, the mean RMSE values range between 10 and 20, with a maximum at more than 38. From this experiment, it can be observed that k-fold cross-validation with blocking will generate a better forecast.

The RMSE values for the different splits from k-fold cross-validation with blocking are shown in Figure 3-a. In general, the performance of the models are quite consistent despite some peaks at k=5 and k=7 values for certain models.

### B. Impact of Principle Component Analysis on Forecast Performance

The effects of PCA on model prediction's RMSE scores are shown in Figure 4. The figure contains the RMSE values of the last split of the k-fold train-test set for IBM's daily open price prediction. For some models, like Bayesian Ridge Regression and Support Vector Regression, there is minimal difference in RMSEs of the predicted results. For other models, the difference is more noticeable, with linear regression having the most different of about 5 in terms of RMSE values. For most models, PCA perform better, except for Bayesian ridge regression, and decision tree regression.

### C. Forecast Result

The RMSEs results of the predictions are presented in Figure 5. Amongst all models, LSTM shows the smallest RMSEs overall at around 3 to 5 for IBM. For regression models, linear regression and Bayesian ridge regression show smaller RMSEs at around 5 for IBM. For the JNJ stock, the RMSEs values are even smaller for all model types. However, for AAPL stock, the performance is much worse than the other two. Random forest regression and support vector regression works better for AAPL, but all models have RMSE values of more than 120. This might be due to inconsistency in technical indicator data or large fluctuations in stock price that the models do not account for. To investigate further, Figure 6 shows the effects for each data split. The models work consistently worse than the other two stocks except for the 0th and the 1st split, which indicates that the cause is movement in stock price for the AAPL are not accounted for in these model, and might not be correlated with past technical indicators.

Figure 7, 8, and 9 show the daily open stock price forecast plots for the seven machine learning models on the most recent k-fold splits for IBM, AAPL, and JNJ stocks respectively.

| | linear_reg | bayesian_ridge_reg | lasso_reg | decision_tree_reg | random_forest_reg | support_vector_reg | LSTM |
|---|---|---|---|---|---|---|---|
| open | 5.238736 | 5.255670 | 9.990838 | 11.401453 | 9.928072 | 6.025175 | 4.416079 |
| close | 5.450626 | 5.466204 | 10.025662 | 10.884061 | 10.369667 | 6.013040 | 4.867978 |
| high | 5.282591 | 5.298958 | 10.037611 | 11.168830 | 10.178002 | 6.049455 | 4.599683 |
| low | 5.580053 | 5.589038 | 9.974871 | 11.252080 | 10.022782 | 6.323850 | 3.873659 |

(a) IBM

| | linear_reg | bayesian_ridge_reg | lasso_reg | decision_tree_reg | random_forest_reg | support_vector_reg | LSTM |
|---|---|---|---|---|---|---|---|
| open | 247.705588 | 247.576714 | 207.088619 | 160.200624 | 158.579838 | 130.059057 | 344.329495 |
| close | 248.452536 | 248.244175 | 207.001639 | 161.034401 | 159.230585 | 130.993021 | 329.303187 |
| high | 242.759030 | 242.887869 | 210.990076 | 162.093246 | 160.239230 | 132.099696 | 291.241962 |
| low | 235.959989 | 236.046037 | 203.075705 | 158.001064 | 157.342771 | 128.864515 | 322.989121 |

(b) AAPL

| | linear_reg | bayesian_ridge_reg | lasso_reg | decision_tree_reg | random_forest_reg | support_vector_reg | LSTM |
|---|---|---|---|---|---|---|---|
| open | 2.327252 | 2.222550 | 3.496379 | 5.783765 | 3.203860 | 2.951002 | 3.253116 |
| close | 2.813179 | 2.625168 | 3.532700 | 5.948018 | 3.767624 | 3.451273 | 3.203796 |
| high | 2.743340 | 2.598636 | 3.416536 | 4.496117 | 3.247503 | 2.965093 | 3.198960 |
| low | 2.196782 | 2.108104 | 3.569166 | 4.785107 | 3.273932 | 3.143420 | 3.489556 |

(c) JNJ

Fig. 5. RMSE values for IBM, AAPL, and JNJ stocks' daily open, daily close, daily high, and daily low prices

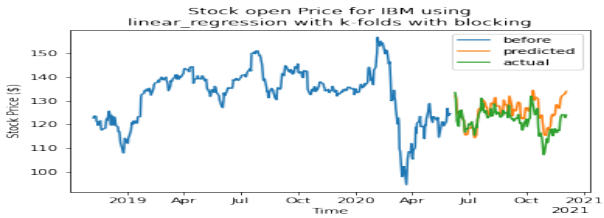
| | linear_reg | bayesian_ridge_reg | lasso_reg | decision_tree_reg | random_forest_reg | support_vector_reg | LSTM |
|---|---|---|---|---|---|---|---|
| 0 | 7.258653 | 6.794513 | 12.861540 | 2.631931 | 2.477904 | 4.347233 | 3.301521 |
| 1 | 0.557950 | 0.561841 | 1.295471 | 1.699493 | 1.423310 | 1.757562 | 0.767702 |
| 2 | 16.562292 | 16.094719 | 11.143329 | 24.753562 | 22.175465 | 33.691573 | 50.612835 |
| 3 | 7.097900 | 7.071044 | 21.425688 | 26.032293 | 19.061884 | 30.447971 | 14.863281 |
| 4 | 4.748757 | 4.746221 | 23.820600 | 40.720026 | 38.437601 | 61.764592 | 16.443504 |
| 5 | 21.534364 | 21.648461 | 74.688038 | 181.258862 | 176.285077 | 226.867004 | 125.229273 |
| 6 | 364.732933 | 365.050357 | 364.129048 | 377.459337 | 373.324684 | 325.480096 | 385.027154 |
| 7 | 2.324398 | 2.327589 | 5.868285 | 5.519971 | 5.174905 | 4.807437 | 4.152857 |
| 8 | 7.543220 | 7.577276 | 18.986991 | 38.840100 | 34.857361 | 45.618264 | 14.040421 |
| 9 | 247.705588 | 247.576714 | 207.088619 | 160.200624 | 158.579838 | 130.059057 | 337.599183 |
| mean | 68.006606 | 67.944874 | 74.130761 | 85.911620 | 83.179803 | 86.484079 | 95.203773 |

Fig. 6. AAPL's RMSE values for k-fold cross validation with blocking

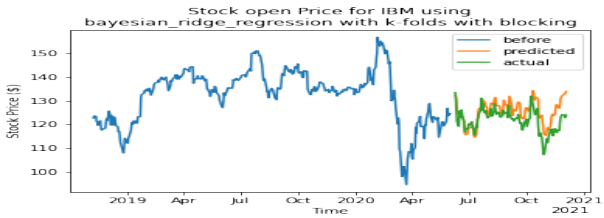
## V. Conclusion and Future Works

Daily stock price are predicted by applying machine learning models to past 5 days' technical indicators, that consists of the volume of trade, dividend, simple moving average (SMA), exponential moving average (EMA), moving average convergence/divergence (MACD), stochastic oscillator (STOCH), relative strength index (RSI), average directional movement index (ADX), commodity channel index (CCI), Aroon indicator (AROON), Bollinger bands (BBANDS), Chaikin A/D line (AD), and on balance volume (OBV). The stocks that were investigated are IBM, Apple and Johnson and Johnson. Stock price and technical indicator data for the past almost 20 years are used to train the models after pre-processing steps such as k-fold cross-validation split with blocking, normalization with standard scaler and min max scaler, as well as principle component analysis for dimensionality reduction.

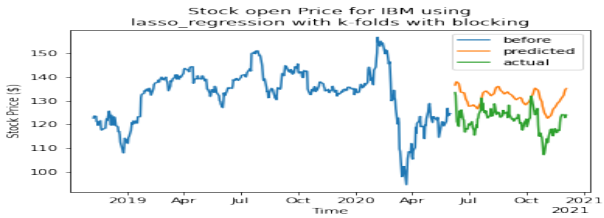
From measuring the root mean square error of the prediction with the actual stock price, the best models are long short term memory model, linear regression model and Bayesian ridge regression model. For IBM stock, they all have a RMSE score of around 2 to 5 for all price types. For JNJ, even smaller RMSEs are obtained. However, this approach of predicting stocks through technical indicators do not seem to work for AAPL. All the models have a RMSEs of above 120. Therefore, this approach of stock price prediction through analyzing technical indicators might not be applicable to all kinds of stocks, but it is promising to some particular stocks that have



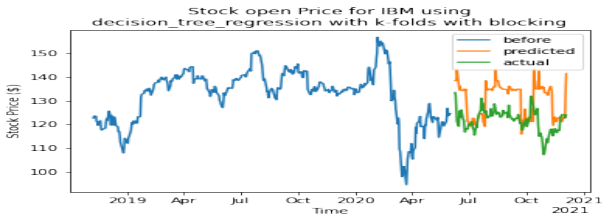

(a) Linear Regression



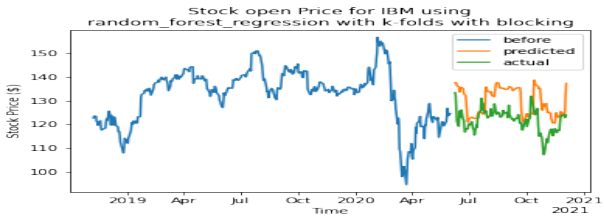

(b) Bayesian Ridge Regression



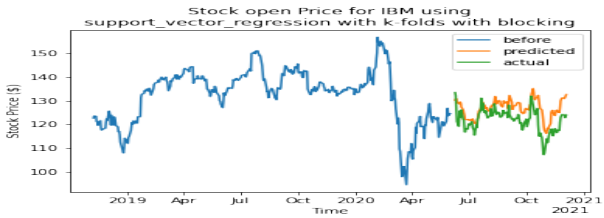

(c) Lasso Regression



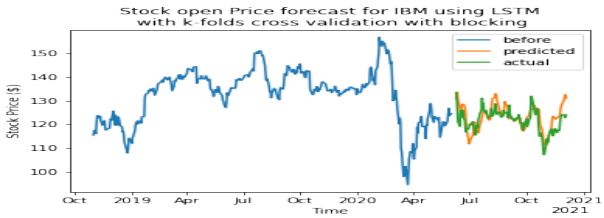

(d) Decision Tree Regression




(e) Random Forest Regression

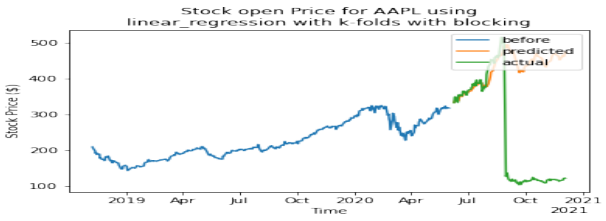



(f) Support Vector Regression

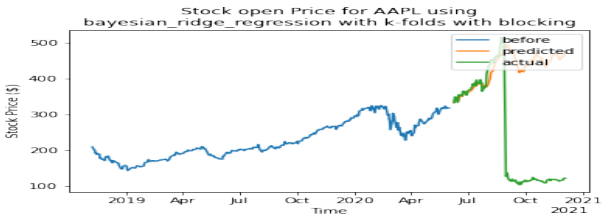



(g) Long Short Term Memory

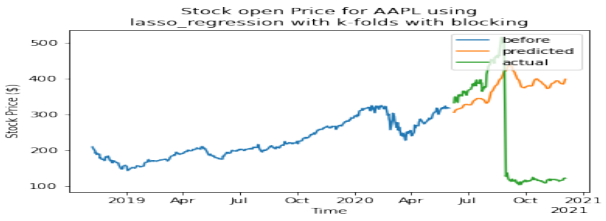
Fig. 7. IBM stock price prediction with seven different machine learning models)



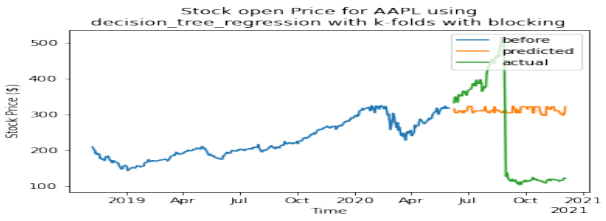

(a) Linear Regression



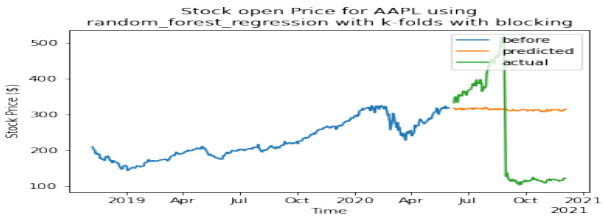

(b) Bayesian Ridge Regression



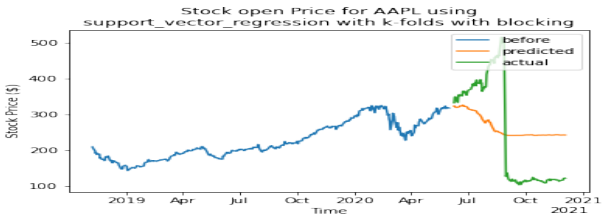

(c) Lasso Regression



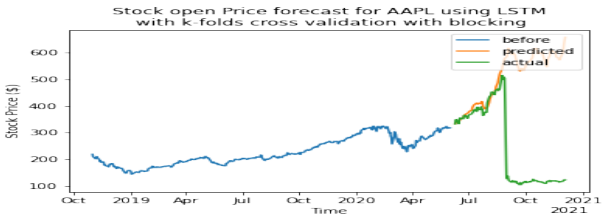

(d) Decision Tree Regression




(e) Random Forest Regression



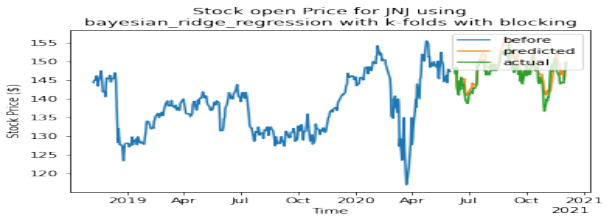

(f) Support Vector Regression




(g) Long Short Term Memory

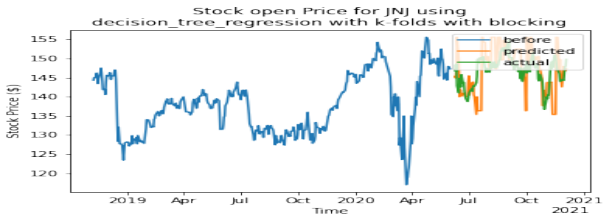
Fig. 8. AAPL stock price prediction with seven different machine learning models)


(a) Linear Regression

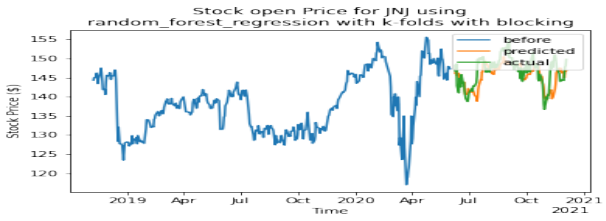



(b) Bayesian Ridge Regression

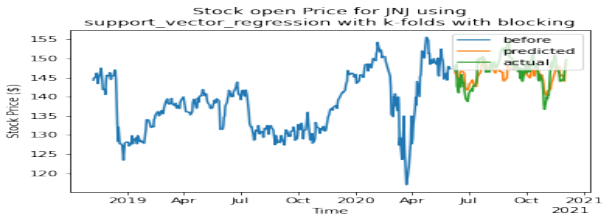

(c) Lasso Regression



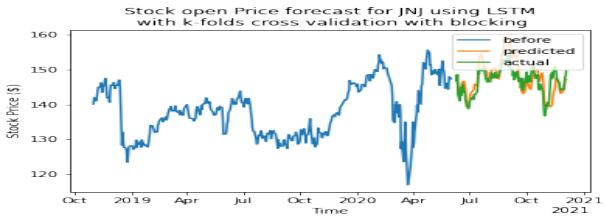

(d) Decision Tree Regression




(e) Random Forest Regression




(f) Support Vector Regression




(g) Long Short Term Memory

Fig. 9. JNJ stock price prediction with seven different machine learning models)

trends more correlated with past technical indicators.

Through designing the machine learning pipeline, the effects of k-fold cross-validation with or without blocking and whether to perform PCA is studied. From results of experimentation, k-fold cross-validation with blocking shows better prediction result. This is consistent with the nature of the stock price data. As the stock market changes, the trend and correlation with technical indicator also changes. Therefore, training a model over a very long period of time by not using blocking will make the model too generalized for future predictions. The impact of PCA on prediction performance are different for each model. In general, the dimensionality reduction produces better result, but the main benefit comes from reduced computation time.

To improve on this project, the independent variable can be adjusted to price difference between next day and current day. This might improve the prediction result, as in this project, actual past stock prices are not considered during prediction. This will help to isolate the effect of original stock price. Another improvement to the project can be to obtain a multi-day forecast instead of just a next-day forecast, by continue predicting on predicted data. However, this will require calculations of technical indicators based on daily open, daily close, daily high, and daily low predicted prices. Since some features depend on volume of trade, it would require some assumption or prediction for volume of trade to obtain in order to build further predictions. One additional feature can be added before using this approach is to identify stocks that are likely to be correlated with technical indicators like the IBM and JNJ studied in this case. This will save time and computational power so it is not wasted to obtain unusable results for stocks like AAPL. One the other hand, more factors like company news and fundamental analysis of the stock can be included to make this approach more generalized for all kinds of stocks.

## BIBLIOGRAPHY

## REFERENCES


[1] W. Jasic, "The profitability of daily stock market indices trades based on neural network predictions: case study for the S&P 500, the DAX, the TOPIX and the FTSE in the period 1965-1999," Applied financial economics, vol. 14, no. 4, pp. 285–297, Feb. 2004, doi: 10.1080/0960310042000201228.

[2] C. Yu, "Evolving Least Squares Support Vector Machines for Stock Market Trend Mining," IEEE transactions on evolutionary computation, vol. 13, no. 1, pp. 87–102, Feb. 2009, doi: 10.1109/TEVC.2008.928176.

[3] D. L. Minh, A. Sadeghi-Niaraki, H. D. Huy, K. Min, H. Moon, Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network, Ieee Access 6 (2018) 55392–55404 (2018).

[4] W. Wang, "Stock2Vec: A Hybrid Deep Learning Framework for Stock Market Prediction with Representation Learning and Temporal Convolutional Network," Sep. 2020.

[5] K. Ntakaris, "Mid-price prediction based on machine learning methods with technical and quantitative indicators," PloS one, vol. 15, no. 6, pp. e0234107–e0234107, Jun. 2020, doi: 10.1371/journal.pone.0234107.

[6] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques," Expert Systems with Applications, vol. 42, no. 1, pp. 259–268, 2015.