Predicting Stock Price Using Supervised Machine Learning

Anirudhan Anbuchezhian

School of Computing and Mathematical Sciences
University of Greenwich
London, United Kingdom
aa0854y@greenwich.ac.uk

***Abstract*—Stock market is an exchange where individual and institutional investors come to buy and sell shares of a public venue. Investments in the stock market tend to be high risk, due to its chaotic nature where the stock unexpectedly goes up and down. This paper shows how a machine learning algorithm can be used to predict the future stock price, which can help investors minimize the risk of their investments. Regression was used to predict stock closing prices. RMSE, R^2, MSE, MAE are used to evaluate the performance of the created model. Overall, these regression models showed promising results in the prediction of the stock closing prices.**

Keywords—Sklearn, Polynomial Regression, Linear Regression, Stock Market, Machine Learning

I. Introduction

The stock market is an exchange where investors come to buy or sell shares to make a profit. The shares are a part of any public company, which are issued through the process of IPO (Initial Public Offerings). The companies sell their shares to the common public to raise capital for their developments. The stock market is highly volatile and there are too many factors to consider which can affect stock prices. Future events are unpredictable due to its chaotic nature. This makes any investments in the stock market a high risk.

The prediction of future prices using advanced knowledge helps minimize the risk of investment. Investors and traders buy a stock when predicting the stock to go higher and sell a stock when predicting the stock to go lower. If the trend of the stock is analyzed, then investors and traders would know where the price is going to head in the future. This is where technology can help with predicting future prices by solving complex equations. Machine learning is one of the tools that can do the job of future prices by training the past data.

In statistics, stock prices can be viewed in a graph to identify the dependent and independent variables and try to establish a relationship between them. Linear regression and polynomial regression use this technique to predict the dependent variable using the independent variable.

In this paper, Short term prediction of closing price is going to reviewed with 4 years of dataset. Two models are created using Linear Regression and Polynomial Regression and comparing the performance of each model to check which model performed better at predicting future prices. Apple stock prices are used as a dataset to train and test the machine learning models. The daily historical data is extracted between 1st January 2016 and 1st January 2020.

The rest of the paper is organized as follows. Chapter II provides a summary of related articles towards the prediction of stock prices using machine learning algorithms. Chapter III shows how and where the data is extracted and how the data preparation is done. Chapter IV describes each machine learning algorithms to create a model and tells how the performance of each model is compared. Chapter V shows the results produced by each model with their respective performance. Chapter VI and VII concludes the paper about the process we have taken to predict future prices and conveys how it can be improved further in the future.

II. Literature Review

Kunal Pahwa and Neha Agarwal ***(Pahwa, K. and Agarwal, N., 2019)*** created a simple machine learning model to make the unpredictable stock market into more predictable. Their main idea was to show how powerful a machine learning algorithm is and how important is the data that is fed to a machine learning algorithm. They used the simplest classifier of supervised machine learning algorithm i.e. Linear Regression, using the Sklearn library. Their model was able to produce acceptable results. Their view for the accuracy of model should not be less than 95%, which intend to be useless since accuracy is a very crucial factor in a machine learning model.

R.Seethalakshmi ***(Seethalakshmi, R., 2018)*** applied Linear Regression to predict the closing price of the S&P 500 by finding the best independent variable for the prediction. She created two different Linear Regression models with different independent variables and compared the performance of both the models using AIC (Akaike Information Criterion), BIC (Bayesian Information Criterion) and R^2 Scores. Her first model had open, high, low, volume and adj.close attributes as the independent variable to predict the closing price. Her second model had open, low, high attributes as the independent variable to predict the closing price. Overall, model 1 with all attributes had better R^2 score and less AIC, BIC values then model 2. She concludes that open, high, low, volume and adj.close were essential for predicting the closing value accurately.

Yumo Xu and Shay B. Cohen ***(Xu, Y. and Cohen, S.B., 2018)*** created a model to predict stock movement using a deep generative model called StockNet. The data applied to this model is tweets and daily historical data. accuracy and MCC (Matthews Correlation Coefficient) is used as an evaluation metrics for the model. This model was compared with five

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

other models which are Rand, ARIMA, Random Forest, TSDLA and HAN. Their StockNet model performed better than other baseline models using social media data and historical data.

Bruno Miranda Henrique, Vinicius Amorim Sobreiro and Herbert Kimura ***(Henrique, B.M., Sobreiro, V.A. and Kimura, H., 2018)*** created a model to predict stock prices for large and small companies in three different markets using support vector regression. Daily and minute data were applied to the model. Performance errors are measured for different models using a random walk. The model was able to predict stock price when the model is updated periodically.

III. Data Preparation

A. Data Collection

The dataset we will be using to create ML (Machine Learning) models is raw historical data downloaded from https://finance.yahoo.com/. Apple stock is going to be used for creating the models. The dataset is downloaded as a CSV format and then loaded as a data frame. The data extracted consists of 5 years of data.

The attributes of the data include:

Date (Date of the stock being traded),
Open (Open price of the stock of the day),
High (Highest price of the stock of the day),
Low (Lowest price of the stock of the day),
Close (Closing price of the stock of the day),
Volume (Total volume traded of the day),
Adj. Close (Adjected Closing price of the stock of the day)

'Close' attribute of the dataset will be the variable that is going to be predicted. For the model, 'Open' attribute is going be the independent variable and 'Close' attribute is going to be the dependent variable.

B. Data Cleaning

Before the dataset is used for ML models, First, we need to deal with all the missing values that may consist in the data present. For this paper, we will be dropping all the rows that have no values present. By doing this will give us an accurate prediction while applying the machine learning algorithm.

IV. Methodology

A. Dataset Partition

Once the data preparation is completed, the dataset is ready to be used for creating Machine Learning models. Before creating machine learning models, the dataset needs to be split into a training set and testing set. The splitting is essential to be able to evaluate the performance of the model. If all of the available data is used for training the models, it would be difficult to predict the unseen data. 80% of the total dataset will be stored as a training set and the rest of the 20% will be stored as a testing set. The training set will have data from 1st January 2016 to 1st April 2019 and the testing set will have data from 1st April 2019 to 1st January 2020.

The model is going to be created by using python programming language in the Jupiter notebook. The function 'train_test_split' will be imported from 'sklearn.model_selection' library. Since this function randomizes the data that is passed to it, shuffle will be switched off by setting it to false. The below figure shows the code, which is used for splitting the dataset into the training and testing set.

```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.2,shuffle=False)
```

Figure 1:Python code for data partition

B. Machine Learning

Since the problem we are facing is to predict future price, regression is going to be used to create the models. Linear Regression and Polynomial Regression are the two machine learning algorithms that are used to create the models. Both of these are supervised machine learning algorithms.

a) Linear Regression

Linear Regression is used for predictive analysis, where a linear line is drawn by modelling the relationship between a dependent variable and an independent variable. The linear model which is created from the equation, is the following:

$$y = \alpha + \beta x$$

β = slope
α = y-intercept
y = y- coordinate
x = x-coordinate

Figure 2:Formula for Linear Regression

Linear Regression will be imported from 'sklearn.linear_model'. This function will use the training set to create a linear regression model, which will be used to predict the dependent variables using independent variables for the testing set. The figure below shows the code used for creating the model as well as predicting the prices using the model.

```
# creating a Linear Regression model
model = LinearRegression()
model.fit(x_train, y_train)
y_pred = model.predict(x_train)
```

Figure 3:Linear Regression model created using python programming language

b) Polynomial Regression

Polynomial Regression is also used for predictive analysis like Linear Regression. Unlike linear Regression, a non-linear line is drawn by modelling the relationship between the dependent variable and nth degree function of the independent variable. The non-linear model, which is created from the equation, is the following:

$$y = b_0 + b_1 x_1 + b_2 x_1^2 + \ldots + b_n x_1^n$$

Figure 4:Formula for Polynomial Regression

Where n is the degree of the polynomial and the b is a set of coefficients.

Polynomial Regression will use the same function as the Linear Regression, which are both imported from 'sklearn.linear_model'. Before this function is used to create the model, polynomial features will be used to set the nth degree to create a non-linear model, which is imported from 'sklearn.preprocessing'. The figure below shows the code that is used to create the model.

```
polynomial_features= PolynomialFeatures(degree=3)
x_test_poly = polynomial_features.fit_transform(x_test)
x_train_poly = polynomial_features.fit_transform(x_train)

model = LinearRegression()
model.fit(x_train_poly, y_train)
y_pred = model.predict(x_train_poly)
```

Figure 5: Polynomial Regression model using python programming language

C. Evaluation Metrics

Since regression is used to create the models, the metrics that will be used to evaluate the accuracy of the model are: R square score, root mean square error (RMSE), mean square error (MSE) and mean absolute error (MAE).

The description of each evaluation metrics is shown below:

1. R-Square

$$R^2 = 1 - \frac{RSS}{TSS}$$

R^2 = coefficient of determination
RSS = sum of squares of residuals
TSS = total sum of squares

Figure 6: Formula for R-Square Metrics

The model is a good fit if the R square score is closer to 1.

2. RMSE

$$RMSE = \sqrt{(f-o)^2}$$

Where:

- f = forecasts (expected values or unknown results),
- o = observed values (known results).

Figure 7: Formula for RMSE Metrics

The model is a good fit if the RMSE score is closer to 0.

3. MSE

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MSE = mean squared error
n = number of data points
Y_i = observed values
$\hat{Y}_i$ = predicted values

Figure 8: Formula for MSE Metrics

The model is a good fit if the MSC score is closer to 0.

4. MAE

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

MAE = mean absolute error
y_i = prediction
x_i = true value
n = total number of data points

Figure 9: Formula for MAE Metrics

The model is a good fit if the MAE score is closer to 0.

V. Results

In this chapter, the results obtained by both Linear and Polynomial Regression models are compared to each other. The performance of the model is tested by using the test dataset after the model is trained using the training dataset.

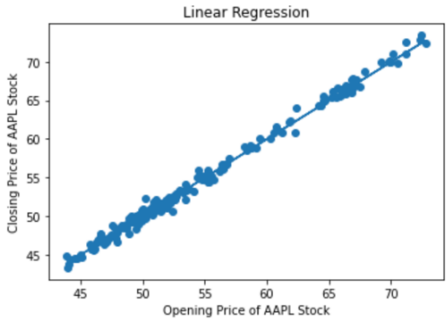

Figure 10:Linear Regression model plotted with scatter points

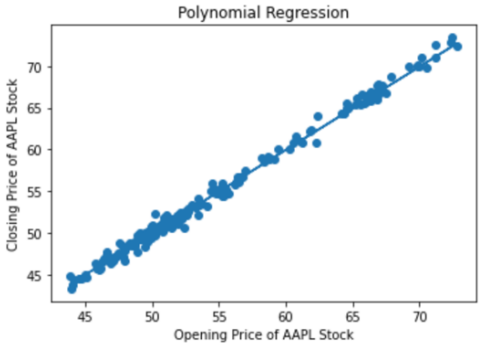

Figure 11: Polynomial Regression model plotted with scatter points

Table 1: Comparing both models with regression metrics

Metrics	Linear Regression Model	Polynomial Regression Model
R-Square	0.99405	0.99389
Root Mean Square Error (RMSE)	0.57452	0.58209

Mean Square Error (MSE)	0.3300	0.33883
Mean Absolute Error (MAE)	0.4418	0.44847

From looking at Table 1, Both models performed well at predicting APPL stock price.

When looking at Metrics:

- R-Square score is closer to 1 for the linear regression model than in the Polynomial Regression model.
- Root Mean Square Error (RMSE) score is closer to 0 for Linear Regression model than in the Polynomial Regression model.
- Mean Square Error (MSE) score is closer to 0 for Linear Regression model than in the Polynomial Regression model.
- Mean Absolute Error (MAE) score is closer to 0 for Linear Regression model than in the Polynomial Regression model.

Overall, looking at the metrics scores, Linear Regression performed better than the Polynomial Regression model when predicting the closing price of apple stock, by using open price as an independent variable and close price as the dependent variable.

VI. Conclusion

In this paper, Linear and Polynomial regression models were created to predict future closing prices in the stock market. Both of the models were able to successfully predict the closing prices for apple stock. From looking at the R^2 score we can see that Linear Regression and Polynomial Regression had a score of 0.9940 and 0.9938, showing how a model performed well. This paper only covers supervised machine learning and instructions on how to create a machine learning model. More advanced machine learning algorithms, such as deep learning and time series forecasting can be applied to further improve the model for the future.

VII. Future Work

The models which were created by using linear regression and polynomial regression are not meant to be used for actual prediction of the stock prices. This is because the opening price was used as an independent variable to predict the closing price of the stock. The purpose of this paper is to show how machine learning models can be created to predict the stock market trend which makes any investments in the market a low risk. The models that have been created in this paper is impossible to use in the actual market because the opening price of the stock is unpredictable as the closing price. Time series forecasting such as Arima model, RNN or regression etc. can be used to predict the closing price of the stock, which is something that can be modified more in the future. By using time series forecasting, the model can predict future values based on previously observed values. Fundamental analysis can also be something that can be looked at to predict stock prices and check whether the stock is undervalued or overvalued based on companies' financial statement.

References

[1] G. E. P. Box and G. M. Jenkins, Time series analysis: forecasting and control. San Fransisco, CA: Holden- Day, 1976. to Bull and Bear Markets, President, Global Financial Data, Inc.

[2] Henrique, B.M., Sobreiro, V.A. and Kimura, H., 2018. Stock price prediction using support vector regression on daily and up to the minute prices. *The Journal of finance and data science*, *4*(3), pp.183-201.

[3] Pahwa, K. and Agarwal, N., 2019, February. Stock Market Analysis using Supervised Machine Learning. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)* (pp. 197-200). IEEE.

[4] Seethalakshmi, R., 2018. Analysis of stock market predictor variables using Linear Regression. *Int. J. Pure Appl. Math*, *119*(15), pp.369-378.

[5] Wulff, S.S., 2017. Time series analysis: Forecasting and control. *Journal of Quality Technology*, *49*(4), p.418.

[6] Xu, Y. and Cohen, S.B., 2018, July. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1970-1979).