



ANALYZING THE STOCK MARKET PRICE USING LINEAR REGRESSION

Shubham Khedkar¹, Vadivu.G²,
Navin.K³

SRM Institute of Science and Technology

M. Tech in Big Data Analytics

shubhamkhedkar777@gmail.com,

vadivukar@gmail.com,

navin.k@ktr.srmuniv.ac.in

July 9, 2018

Abstract

Machine Learning is used in many data analytics problems to predict the future with more accuracy. In stock market, the prediction is more difficult because of the nature of frequent changes in the incoming dataset. Machines are not having the knowledge by itself; it has to be trained explicitly to make right decisions. When a machine improves its performance based on its past experiences it can be said that machine has truly learnt. The technique for most accurate prediction is by learning from past instances, and to make a program to do this is best possible with machine learning techniques. The main objective of this paper is to predict the market performance of Indian Stock Exchange based on day closing values using different machine learning techniques. By using different attributes as an input prediction is done to identify Positive or Negative value. Linear Regression algorithm is used to predict the market in this work. As a result while predicting the market performance, predicting the attribute which is contributing more to predict the

market performance is also identified in this work. The result proves that the performance of Indian stock market exchange can be predicted with machine learning techniques.

Keywords: Machine Learning, Linear Regression, Big Data, Prediction

1 Introduction

To address this, a dataset which contains different attributes like date, High value, Low value, closing price, volume are considered. Dataset has been created from the observations available online. Linear regression algorithm is used in this research paper. With the usage of regression algorithm, it is possible to predict stock price and decision can be taken whether to buy the product or not.

Stock market is one of the first options of attracting investments and main financial indexes of country. Stock trading volume includes the number of lots bought and sold which is expressing in daily basis. The more trading volume of a stock is higher, the more the stock is active. Trading volume is an approving to price patterns in technical analysis and it's more important than stock price. People could predict the moving direction of trading volume of a stock in the future, can also obtain the prices changing, continuing or finishing of its trend, with more confidence.

Nowadays the stock market has been called for research in many fields due to its effects on financial challenging and capacity of predicting its various aspects through different scientific methods such as genetic algorithm, Artificial Neural Network (ANN) and other Meta heuristic algorithms. Stock market price prediction is a problem that has the potential to be worth billions of dollars and is actively researched by the largest financial corporations in the world.

The stock market is complex adaptive system that is difficult to predict due to the large number of factors that determine the day to day price changes. Machine learning through regression which tries to determine the relationship between a dependent variable and one or more independent variables. Here, the independent variables are the features and the dependent variables are the variables which

would like to predict the price.

Python programming language: Python is a widely used general-purpose, high level programming language. There are two major Python versions- Python 2 and Python 3. Both are quite different. For this analysis, SciPy stack with pandas for data wrangling and matplotlib for visualisation are used. Anaconda already comes with all these packages those are needed. Python codes are easier to maintain and more robust than R. Years ago; Python didn't have many data analysis and machine learning libraries. Recently, Python is catching up and provides cutting-edge API for machine learning or Artificial Intelligence. Most of the data science job can be done with five Python libraries: Numpy, Pandas, Scipy, Scikit-learn and Seaborn.

Framework:

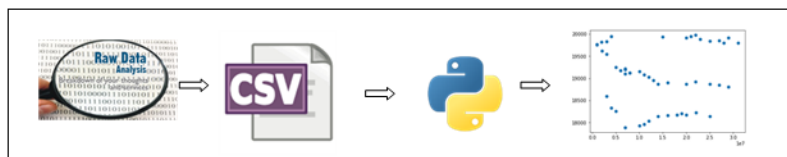


Figure 1: Stock Market Price Prediction Framework Using Linear Regression

Fig.1 shows the process of stock market price prediction using Linear Regression algorithm. Raw data is taken as input, converted into the form of csv. Python language with scikit library is used to apply Linear Regression algorithm on the csv file.

Algorithm: Linear Regression algorithm

Linear regression is used to find relationship between scalar dependent variable Y and one and more explanatory variable denoted as X . In this case one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple regressions. The procedure for linear regression is different and simpler than that for multiple linear regressions. By using simple linear regression on training model learned the relationship in the data.

With simple linear regression we want to model our data as follows:

$$Y = B_0 + B_1 * X$$

Linear regression gives best fit line as output. In this Y is the output variable as predicted variable based on X as the input variable, B0 and B1 are coefficient.

Coding:

Training and Testing model:

The below code is used to split the dataset into training and test dataset. The input dataset is taken from the online and converted into comma separated file (csv) format. Pandas is the library which is imported to find the dataset in dataframe format. Python has long been great for data preparation, but less useful for data analysis and modeling. Pandas helps to fill this gap, enabling to carry out the entire data analysis workflow in Python. Dataframe is the two-dimensional size mutable, potentially heterogeneous tabular data structure with labelled rows and columns. Arithmetic operations align on both row and column labels.

Matplotlib is a plotting library for the Python programming language and its numerical library extension NumPy. It provides an object-oriented API for plotting plots into applications using general-purpose GUI.

Linear regression analysis is the most widely used among many available statistical techniques. It is the study of linear, additive relationships between variables. Let Y denote the dependent variable whose values you wish to predict, and let X1, ,Xk denote the independent variables from which one can predict, with the value of variable Xi in period t, denoted by Xit. Then the equation for computing the predicted value of Yt is:

$$\hat{Y}_t = b_0 + b_1 X_{1t} + b_2 X_{2t} + \dots + b_k X_{kt}$$

This formula has the property that the prediction for Y is a straight-line function of each of the X variables, considering the others as fixed, and the contributions of different X variables to the

predictions are additive. The slopes of their individual straight-line relationships with Y are the constants $b_1, b_2, \dots, b_k$, they are called as coefficients of the variables. b_i is the change in the predicted value of Y per unit of change in X_i , other things are equal. The additional constant b_0 , the called as intercept, is the prediction that the model would make if all the X 's were zero. The coefficients and intercept are estimated by least squares, setting them equal to the unique values that minimize the sum of squared errors within the sample of data to which the model is fitted. The model's prediction errors are typically assumed to be independently and identically normally distributed.

```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
%matplotlib inline
da = pd.read_csv("stock.csv")
da.head()
x = da.iloc[:, :-7].values
y = da.iloc[:, -4].values
from sklearn.cross_validation import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(x, y, random_state=0)
# splitting the rows into training and test data sets
print(Y_test)
print(X_test)
plt.plot(x, y, 'o')
```

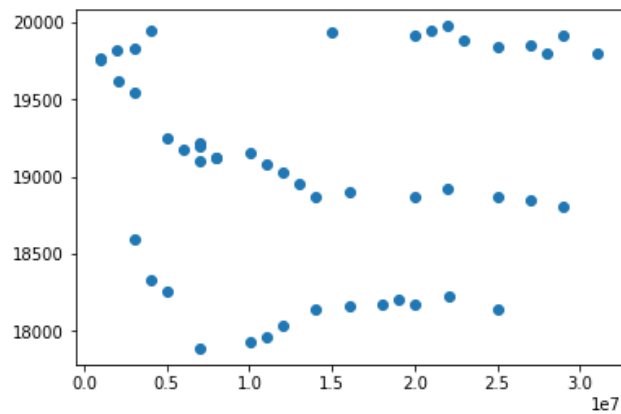


Figure 2: Linear Regression output

Fig.2 shows the correlation plot of test values and their relationship with output values.

```

From sklearn.cross_validation import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(x, y, random_state=0)
from sklearn.linear_model import LinearRegression
lr = LinearRegression().fit(X_train, Y_train)
y_predict = lr.predict(X_test)
plt.scatter(X_train, Y_train, color="red")
plt.plot(X_train, lr.predict(X_train))

```

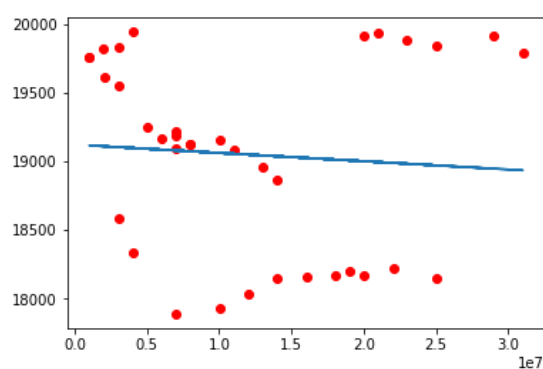


Figure 3: Linear Regression output for Training and Prediction

```

y_predict
print("Training set score: {:.2f}".format(lr.score(X_train, Y_train)))
print("Test set score: {:.2f}".format(lr.score(X_test, Y_test)))

```

output:

Training set score: 0.01
 Test set score: 0.09

Prediction: import csv
 import numpy as np
 from sklearn import linear_model
 import matplotlib.pyplot as plt

```

dates = []
prices = []

```

```

def get_data(filename):
    with open(filename, 'r') as csvfile:
        csvFileReader = csv.reader(csvfile)
        next(csvFileReader)
        for row in csvFileReader:
            dates.append(int(row[0].split('-')[0]))
            prices.append(float(row[1]))
        return

```

```

def predict_price(dates, prices, x):
    dates = np.reshape(dates, (len(dates),1)) # converting to matrix
    of n X 1
    prices = np.reshape(prices, (len(prices),1))

```

```

linear_mod =
linear_model.LinearRegression() # defining the linear regression
model
linear_mod.fit(dates, prices) # fitting the data points in the model

```

```

plt.scatter(dates, prices, color= 'black', label= 'Data') # plotting
the initial datapoints
plt.plot(dates, linear_mod.predict(dates),

```

```

color= 'red', label= 'Linear model')
# plotting the line made by linear regression
plt.xlabel('Date')
plt.ylabel('Price')
plt.title('Linear Regression')
plt.legend()
plt.show()

return linear_mod.predict(x)[0][0],
linear_mod.coef_[0][0],
linear_mod.intercept_[0]

get_data('stock.csv') # calling get_data method
by passing the csv file to it
print("Dates- ", dates)
print("Prices- ", prices)

predicted_price, coefficient, constant = predict_price(dates, prices,
29)
print("\n The stock open price for 30th March is: $",
str(predicted_price))
print("The regression coefficient is ",
str(coefficient), ", and the constant is ",
str(constant))
print("the relationship equation between dates and prices is: price
= ", str(coefficient), "* date + ", str(constant))

```

Output:

```

Dates- [1012016, 2012016, 3012016, 4012016, 15012016,
20012016, 21012016, 22012016, 23012016, 25012016, 27012016,
28012016, 29012016, 31012016, 1022016, 2022016, 3022016,
5022016, 7022016, 6022016, 7022016, 8022016, 8022016, 7022016,
10022016, 11022016, 12022016, 13022016, 14022016, 16022016,
20022016, 22022016, 25022016, 27022016, 29022016, 3032016,
4032016, 5032016, 7032016, 10032016, 11032016, 12032016,
14032016, 16032016, 18032016, 19032016, 20032016, 22032016,
25032016]

```

```

Prices- [19852.55078, 19878.43945, 19981.10938, 19980.24023,
19934.15039, 19933.83008, 19986.56055, 19987.63086, 19917.7793,

```

19923.16992, 19951.28906, 19966.42969, 19953.75, 19824.58984, 19757.74023, 19664.9707, 19558.41992, 19255.89063, 19274.84961, 19196.14063, 19214.30078, 19225.28906, 19144.40039, 19138.7207, 19152.14063, 19083.75977, 19043.90039, 18960.75977, 18915.74023, 18904.0293, 18909.84961, 18925.25977, 18934.05078, 18855.7793, 18873.66016, 18650.06055, 18400.5, 18263.30078, 17986.75977, 18006.96094, 18044.15039, 18177.00977, 18193.67969, 18257.06055, 18253.19922, 18236.03906, 18241.68945, 18275.03906, 18168.75977]

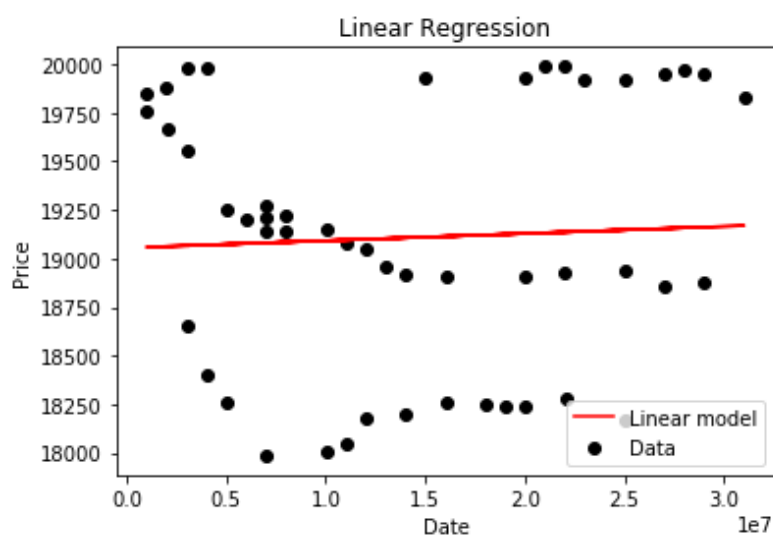


Figure 4: Relationship equation between Dates and Prices

The stock open price for 30th March is: \$ 19054.6129768

The regression coefficient is $3.67447287714 \times 10^{-6}$, and the constant is 19054.6128702 the relationship equation between dates and prices is: $\text{price} = 3.67447287714 \times 10^{-6} * \text{date} + 19054.6128702$

2 Result

The results for 50 day price predictions were chosen to provide insight onto the performance of these algorithms for a longer period of time. Linear regression performed very poorly when its window

size was small for long-term price prediction, but actually ended up outperforming.

3 Conclusion:

Looking at the data, the predictions are quite close, may be not tradable but this gives us a direction. Further improvement in this method includes by adding more than one independent variables and by increasing records in the datasets. Doing so will help reduce the residual or error and help to get closer to the actual price.

References

- [1] C. Tsai, and S. Wang, Stock Price Forecasting by Hybrid Machine Learning Techniques.Proceedings of the International MultiConference of Engineers and Computer Scientists, 20-26, 2009.
- [2] A. S. Chen, M. T. Leung, and H. Daouk, “Application of neural networks to an emerging financial market: forecasting and trading the Taiwan Stock Index”,Computers & Operations Research, 2003, vol. 30, pp. 901-923.
- [3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825-2830, 2011.
- [4] R Vidhya, G Vadivu, Research Document Search using Elastic Search, Indian Journal of Science and Technology 9 (37), 2016.
- [5] T. Y. J. Naga Malleswari , G. Vadivu, “Map reduce: A Technical Review”, Indian Journal of Science and Technology, Vol 9(1), January 2016.
- [6] K.Sornalakshmi, G.Vadivu, “A Survey on Realtime Analytics Framework for Smart Grid Energy Management”,

International Journal of Innovative Research in Science, Engineering and Technology, March 2015.

- [7] Wang, Yanshan, and In-Chan Choi. 2013. "Market Index and Stock Price Direction Prediction using Machine Learning Techniques: An empirical study on the KOSPI and HSI." ScienceDirect (ScienceDirect) 1-13.
- [8] Yan, Robert, and Charles Ling. 2007. "Machine Learning for Stock Selection." Industrial and Government Track Short Paper Collection 1038-1042.
- [9] Navin, G. Vadivu, "Big Data Analytics for Gold Price Forecasting Based on Decision Tree Algorithm and Support Vector Regression (SVR)", International Journal of Science and Research (IJSR), March 2015.
- [10] Hemavathi.D Sornalakshmi.K, Vadivu.G, Sujatha.G, "A survey on using social media data analytics for pharmacovigilance", Research Journal of Pharmacy and Technology, Volume 10, 2017.
- [11] Akshaya.S, Archana.M, Vadivu.G, "Predicting Bus Arrival Time Using Data Analytics", International Journal of Pure and Applied Mathematics, Volume 117, 2017.

