Stock Market Prediction

Parth Joshi (1001556783)
Department of Computer Science
University Of Texas, Arlington
parthkiran.joshi@mavs.uta.edu

Vrushali Kadam (1001514762)
Department of Computer Science
University Of Texas, Arlington
vrushali.kadam@mavs.uta.edu

Mohana Upale (1001556744)
Department of Computer Science
University Of Texas, Arlington
Mohanaajit.upale@mavs.uta.edu

Abstract

The aim of the project is to predict the future stock returns based on the available historical data from Quandl for companies like Google and Facebook. We do this by implementing time series forecasting using ARIMA model.

1 Introduction

The fluctuation of Stock Market is violent and there are many complicated financial indicators. However, with the advent of newer and better technologies, we now have the opportunity to get a steady fortune from the Stock Market and also can help experts to find out the most informative indicators to make better prediction. The prediction of the market value is of paramount importance in order to maximize the profit of the stock option that needs to be purchased while also keeping in mind the risk factor that need to be considered while buy the stock which should be kept low.

The next section of the paper will be the method where we will explain each process including the dataset in detail. After that we will have a brief outlook of the experiments and some reasoning about the results that we have obtained in the evaluation section. Finally, we provide our verdict on the project in the conclusion section including the related and further work that can be done in similar domain.

2 Method

2.1 Data Set

The data set used is collected from Quandl and contains historic data of the <LIST OF COMPANIES> for the range: From May 18th, 2015 to March 8th, 2016.

The features selected for these companies are as follows:

Feature name	Feature datatype
date	object
open	float64
high	float64
low	float64

close	float64
volume	float64
Exdividend	float64
SplitRatio	float64
AdjOpen	float64
AdjHigh	float64
AdjLow	float64
AdjClose	float64
AdjVolume	float64

2.2 Model Used

Use of ARIMA models for time series analysis to predict the stock prices for <LIST OF COMPANIES>.

2.2.1 ARIMA Model

An **autoregressive integrated moving average (ARIMA)** model is a generalization of an autoregressive moving average (ARMA) model. Both of these models are fitted to time series data either to better understand the data or to predict future points in the series (forecasting). ARIMA models are applied in some cases where data show evidence of non-stationarity, where an initial differencing step (corresponding to the "integrated" part of the model) can be applied one or more times to eliminate the non-stationarity.

The AR part of ARIMA indicates that the evolving variable of interest is regressed on its own lagged (i.e., prior) values. The notion AR(p) indicates an autoregressive model of order p and is defined as follows:

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t$$

where ρ_1,…. ρ_p are the parameters of the model, c is the constant, and ε_t is white noise.

The MA part indicates that the regression error is actually a linear combination of error terms whose values occurred contemporaneously and at various times in the past. In time series analysis, the **moving-average** (**MA**) **model** is a common approach for modeling univariate time series. The moving-average model specifies that the output variable depends linearly on the current and various past values of a stochastic term.

$$X_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}$$

where μ is the mean of the series, the θ_1, ..., θ_q are the parameters of the model and the ε_t, ε_{t-1},..., ε_{t-q} are white noise error terms. The value of *q* is called the order of the MA model.

The I (for "integrated") indicates that the data values have been replaced with the difference between their values and the previous values (and this differencing process may have been performed more than once).

NOTE: ARIMA (p, d, q) means that it you are describing some response variable (Y) by combining a p^{th} order Auto-Regressive model and a q^{th} order Moving Average model. A good way to think about it is (AR, I, MA). This makes your model look the following, in simple terms:

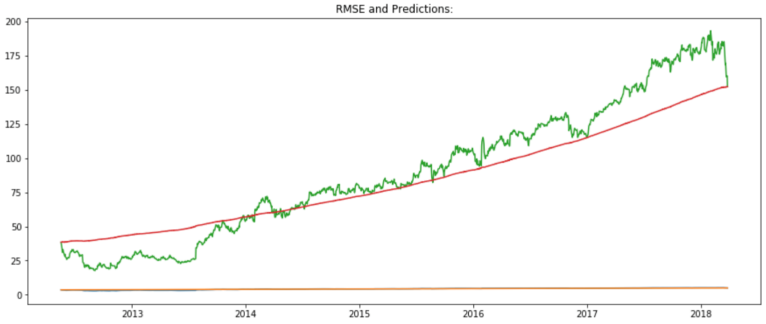
Y = (Auto-Regressive Parameters) + (Moving Average Parameters)

2.3 Methodology

(A) **Data gathering**: Stock quotes for a list of companies are collected by accessing the Quandl repository using Pandas DataReader. The data is loaded in a DataFrame; for each company selected a separate CSV is generated.

(B) **Data conversion and smoothening**: The "Adjacent closing" value for each stock is retrieved and then converted to a time series for further regression. The closing value is selected since the model will predict the prices based on the closing value for the next day. Analysis of trend, seasonality, and residuals is performed using decomposition techniques. Since our model is based on ARIMA model, non-stationary data is smoothened and fitted to become stationary data. Dickey Fuller test is implemented to test the null hypothesis that a unit root is present in an autoregressive model. The alternative hypothesis is different depending on which version of the test is used, but is usually stationarity or trend-stationarity. Difference log transform data to make data stationary on both mean and variance.

DATA ANALYSIS: Trend, seasonality and residual analysis.



(C) **Autocorrelating and choosing model order**: The AFC(Autocorrelation Function) and PAFC(Partial Autocorrelation Function) plots are used to determine the Moving Average(q) and the Autocorrelation Model(p) respectively.

AFC plot: This plot helped us decide the parameters for the AR(p) that is the Autoregression Part of the ARIMA model. The AR model is implemented by passing the parameter to the ARIMA mode as follows:

```
model = ARIMA(timeseries_log, order=(2, 1, 0))
```

PAFC plot: This plot helped us decide the parameters for the MA(q) that is the moving average of the ARIMA model. The MA model is implemented by passing the parameter to the ARIMA mode as follows:

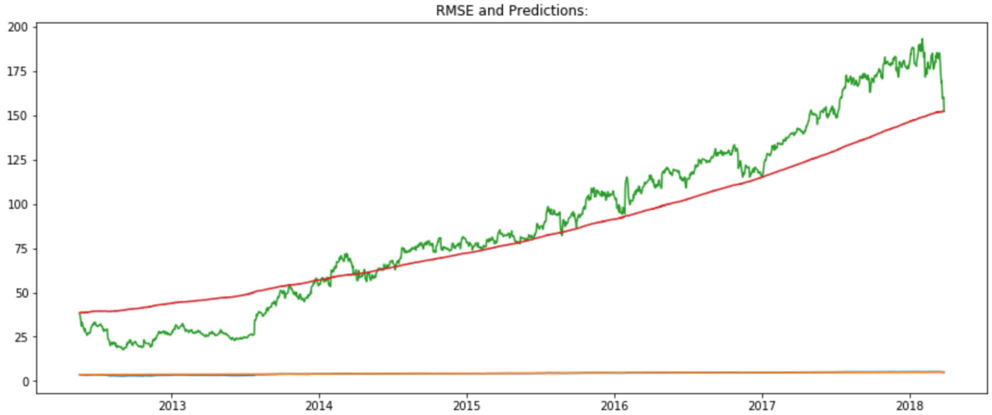
```
model = ARIMA(timeseries_log, order=(0, 1, 2))
```

(D) **Forecasting**: Forecasting the stock prices is done based on the ARIMA parameters selected above. The Autoregression Model correlated the stock prices in the form of a data series to itself and measures how strongly the data values at a specific number of periods apart are correlated to each other over the time. The number of the periods apart is usually called the lag. Our model uses a lag value of 1 which indicates how values one period apart are correlated to one another throughout the series. The Moving average calculate the change in values over a specific time period. The integration is the value of the parameter "d" is set to 1 since the data is converted from non-stationary to stationary series.

(E) **OUTPUT:** In the Output section of the experiment, we have tried to do as much analysis by running dickey fuller on the dataset and plotting various types of analysis like the running average, trend analysis before applying the appropriate the ARIMA model as previously explained. And finally we have applied the ARIMA model and then after applying the prediction model, we have calculated result and have also tried to plot the actuals as well as the predictions so that we can visualize the result and accuracy.

3 Evaluation

In the evaluation, we will be giving a few insights that we found after running the experiment for a few times. From the plots and the dataset, we observed that for any type of company, there is always an overall upward trend in the data. After optimizing the model to perform at its best, we have achieved an accuracy of approximately 70%. Even when we know that the nature of stock markets is very unpredictable but 70% is a good accuracy. The accuracy plot can be briefly seen as follows :



4 Conclusion

In the conclusion section, I would like to say that ARIMA Model is a decent way to predict the Stock Market and the results that we have found prove the same. Having said that, there are other methods also that can be used for doing the same like using tensor-flow, RNNs, linear Regression, etc. In the current scope of implementation we have not included any sentiments about the companies as well as the users. This can be assumed to be a future scope where we are taking into consideration sentiment analysis for the companies as well as the users previous transactions in order to make a better prediction towards the stocks that he might be able to buy. We can also reverse engineer the entire algorithm so that it can give us a particular set of people who might be interested in buying such stock which can be used by stock broking organizations as a head start for targeting.

References

[1] https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/
[2] https://www.youtube.com/watch?v=0xHf-SJ9Z9U
[3] https://www.youtube.com/watch?v=QDrmpphIfLE
[4] http://www.ecostat.unical.it/tarsitano/didattica/SeStoCor/SeStor%2027.3/08notes5GOOD.pdf
[5] https://blog.quandl.com/getting-started-with-the-quandl-api
[6] http://www.forecastingsolutions.com/arima.html
[7] https://people.duke.edu/~rnau/411arim.htm
[8] https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average