Sequence Data Analysis

Harika Kudarvalli
Department of Computer Science
Lakehead University
Thunder Bay, Canada
hkudarva@lakeheadu.ca

***Abstract*—The main goal of the research is to calculate the effective window size of stock's financial status using Linear Regression. For each experiment different type of inputs have been considered. There are six types of variables in the taken dataset. Multiple window sizes are taken and observations have been tabulated and depicted in graphs. The dataset being used consists of the historical prices and volumes of the Nvidia Corporation (nvda ticker),Yahoo Finance Interface for trading days from 2010-Jan31 through the 2018-Dec-31.The results of the observation have been compared against there coefficients, medium absolute error and variance score.**

***Index Terms*—Linear Regression, NVDA, Sequence Analysis, Window Size**

I. Introduction

Linear regression is a simple type predictive analysis. The overall idea of regression is to examine two things: (1) To predict the outcome (dependent variable) of a set of predictor variables is good. (2) Which variables in particular are significant predictors of the outcome variable, and in what way do theyindicated by the magnitude and sign of the beta estimatesimpact the outcome variable? These regression estimates are used to explain the relationship between one dependent variable and one or more independent variables.

Linear regression is the basis for many analyses. Sometimes the data need to be transformed to meet the requirements of the analysis, or allowance has to be made for excessive uncertainty in the X variable. If the requirements for linear regression analysis are not met, alternative robust non-parametric methods can be used. In some data sets, the straight line passes through the origin at 0,0, and then simplified equations can be used. Linear regression is usually used to predict the value of the Y variate at any value of the X variate, but sometimes the inverse prediction is needed, based on a different approach.

Given a time series, predicting the next value is a problem that fascinated a lot of programmers for a long time. Obviously, a key reason for this attention is stock markets, which promised untold riches if you can crack it. However, except for few, those riches have proved elusive.

Thanks to IoT (Internet of Things), time series analysis is poised to make a come back into the limelight. IoT let us place ubiquitous sensors everywhere, collect data, and act on that data. IoT devices collect data through time and resulting data are almost always time series data.

Following are few use cases[1] for time series prediction.

1) Power load prediction
2) Demand prediction for Retail Stores
3) Services (e.g. airline check-in counters, government offices) client prediction
4) Revenue forecasts
5) ICU care vital monitoring
6) Yield and crop prediction

The dataset of NVIDIA organization's stock costs has been extricated from online stock costs dataset. It comprises of 6 sections to be specific, Open, High, Low, Close, Volume and Adjusted Close dependent on the various occasions of the market for example start time and close time.

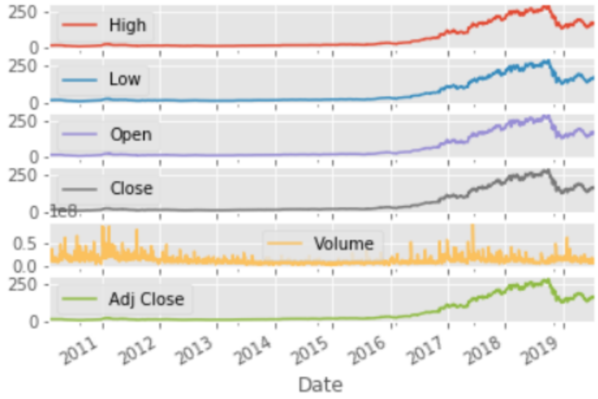


Fig. 1. Plot graph of NVDA observations

II. Implementation

To successfully conduct these experiments, multiple libraries have been used. The Pandas datareader is a sub package that allows one to create a dataframe from various internet datasources, currently including: Yahoo! Finance, Google Finance, St.Louis FED (FRED), Kenneth Frenchs data library, World Bank, Google Analytics. The first step is to create linear regression object. Libraries such as SciKit handles all the math behind all these algorithms. These libraries make lives easier by just exposing APIs, so that we can call those APIs and get our results. We can tune the models, but it is always better to have the knowledge of how a model works internally, so that if required, you can tune it better for your requirements.

Next, fit the model with respect the to data, in other words, the linear regression needs to learn using the training data. Once the model is trained with the training set created, start testing the model with the testing dataset.

The next step is to see how well the prediction is working. For this, use the MatPlotLib library. First, plot the actual values from the dataset against the predicted values for the training set. This will tell how accurate the model is. Then make another plot with the test set. The final step is to evaluate the performance of the algorithm. This step is particularly important to compare how well different algorithms perform on a particular dataset. For regression algorithms, three evaluation metrics are commonly used:

1) Mean Absolute Error (MAE) is the mean of the absolute value of the errors. It is calculated as:

$$\text{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - y_j|$$

Fig. 2. Mean Absolute Error

2) Mean Squared Error (MSE) is the mean of the squared errors and is calculated as:

$$MSE = \frac{1}{N}\sum_{i}^{n}(Y_i - y_i)^2$$

Fig. 3. Mean Squared Error

3) Root Mean Squared Error (RMSE) is the square root of the mean of the squared errors:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

Fig. 4. Root Mean Squared Error

III. Results

Once linear regression is applied to NVDA dataset, following observations can be observed. These observations have been tabulated with their corresponding Median Absolute Error and Variance Score of model - Volume in Table 1.

Window Size	Median Absolute Error	Variance Score
1	2701911.572275	0.562432
5	2696567.993503	0.459297
8	2488084.518700	0.541261
10	2527816.208604	0.402326
14	2425899.603541	0.464364
17	2523495.928651	0.620846
20	2239524.285928	0.399880

Table 1. Observations of Model - Volume

Observations have been tabulated with their corresponding Median Absolute Error and Variance Score of model - Open in Table 2.

Window Size	Median Absolute Error	Variance Score
1	1.204532	0.998763
5	1.608764	0.998165
8	1.484291	0.999239
10	1.109642	0.999691
14	1.164285	0.998094
17	1.707865	0.997720
20	1.565432	0.997863

Table 2. Observations of Model - Open

Predicted and actual values have been plotted in the following graphs. Fig. 5 depicts the values among model - Volume. The actual values are depicted as green whereas predicted values have been shown in blue. As there is less fluctuations between values, the graph has multiple variations.

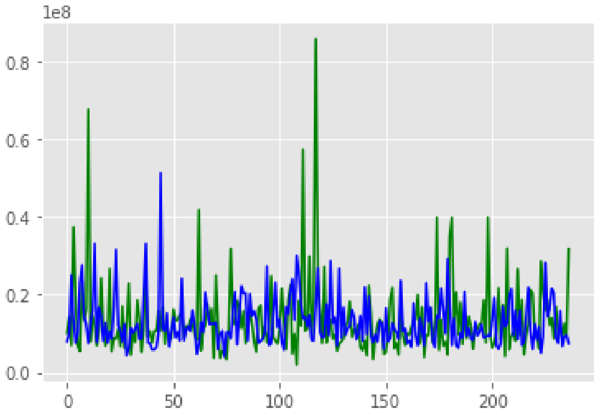


Fig. 5. Plot graph of Model - Volume

Fig. 6 depicts the values among model - Open. The actual values are depicted in red whereas the predicted values are in blue. By understanding the graph it is clear that since there is less fluctuations among values, the predicted and actual values are almost same giving excellent precision score.

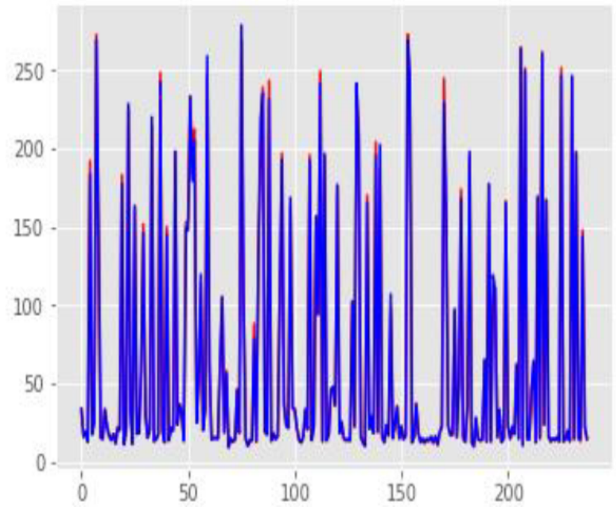
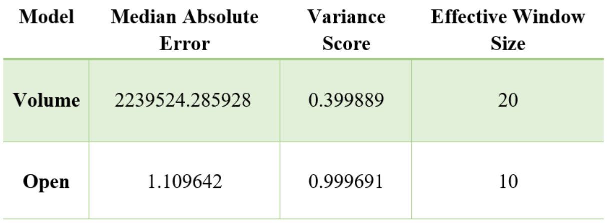


Fig. 6. Plot graph of Model - Open

IV. Conclusion

Linear regression is an extremely simple method. It is very easy and intuitive to use and understand. A person with only the knowledge of high school mathematics can understand and use it. We have successfully applied linear regression to the give NVDA dataset to compare predicted and actual values. From the experiments conducted on Model - Volume, the effective window size is 20 as the error is least and variance score is at 39%. Whereas in Model - Open, the effective window size seems to be 10 with 99% variance score. Model - Open seems to have almost perfect model using linear regression.



Model	Median Absolute Error	Variance Score	Effective Window Size
Volume	2239524.285928	0.399889	20
Open	1.109642	0.999691	10

Table 3. Effective Window Size of different models

References

[1] Sunny Srinidhi, ”Linear Regression in Python using SciKit Learn”.

[2] Nagesh Singh Chauhan, ”A beginners guide to Linear Regression in Python with Scikit-Learn”.

[3] Julien I.E. Hoffman, ”Linear Regression Analysis”.