

Stock Price Prediction: Using an Incremental Learning Approach of Multivariate Linear Regression

Submitted By

Mahtab, Md. Tanvir 17-35246-2

Mahin, Montasir Rahman 17-34978-2

Mia Mohammad Nazim 17-35212-2

Islam, Md. Tanjirul 17-34764-2

A thesis submitted in partial fulfilment of the requirements for the degree of

Bachelor of Science in Computer Science and Engineering

Department of Computer Science

Faculty of Science and Technology

American International University-Bangladesh

June 2021

**Declaration**

We declare that this thesis is our original work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | Mahtab, Md. Tanvir  17-35246-2  Computer Science and Engineering | | |  | | --- | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | Mahin, Montasir Rahman  17-34978-2  Computer Science and Engineering | |
| |  | | --- | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | Mia Mohammad Nazim  17-35212-2  Computer Science and Engineering | | |  | | --- | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | Islam, Md. Tanjirul  17-34764-2  Computer Science and Engineering | |

**Approval**

The thesis titled “Performance analysis of Linear Regression in the Prediction of Stock Prices: An Incremental Learning Approach” has been submitted to the following respected members of the board of examiners of the department of computer science in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering on 22nd June, 2021 and has been accepted as satisfactory.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_­\_\_\_ | | **A.G.M. Zaman**  Assistant Professor & Supervisor  Department of Computer Science  American International University-Bangladesh | | |  | | --- | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_­\_\_\_ | | **Tohedul Islam**  Assistant Professor & External  Department of Computer Science  American International University-Bangladesh | |
| |  | | --- | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_­\_\_\_ | | **Dr. Md. Mahbub Chowdhury Mishu**  Assistant Professor & Head (Undergraduate)  Department of Computer Science  American International University-Bangladesh | | |  | | --- | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_­\_\_\_ | | **Professor Dr. Tafazzal Hossain**  Dean  Faculty of Science & Technology  American International University-Bangladesh | |
| |  | | --- | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_­\_\_\_ | | **Dr. Carmen Z. Lamagna**  Vice Chancellor  American International University-Bangladesh | | |

**Acknowledgement**

Throughout the entire period of thesis we got help from many people and we are indebted to them. We would like to express our gratitude to our supervisor A.G.M. Zaman, Assistant Professor, Department of Computer Science, American International University-Bangladesh for his guidance and encouragement from the very beginning till the end of the thesis. We are also grateful to our external supervisor Tohedul Islam, Assistant Professor, Department of Computer Science, for his valuable instructions.

We would also like to take the opportunity to thank Dr. Carmen Z. Lamagna, Vice Chancellor, and Professor Dr. Tafazzal Hossain, Dean, Faculty of Science & Technology, and Dr. Md. Mahbub Chowdhury Mishu, Assistant Professor & Head (Undergraduate), Department of Computer Science, American International University- Bangladesh, for their support, generosity and encouragement.

Finally we are grateful to our beloved family. Without their support and cooperation the completion of this thesis wouldn’t be possible.

**Table of Contents**

[**Abstract** 1](#_Toc75288221)

[**Chapter 1: Introduction** 2](#_Toc75288222)

[**1.1 Introduction** 2](#_Toc75288223)

[**1.2 Problem Statement** 2](#_Toc75288224)

[**1.3 Scope of the Research** 2](#_Toc75288225)

[**1.4 Objective of the Research** 3](#_Toc75288226)

[**1.5 Significance of the Research** 3](#_Toc75288227)

[**1.6 Research Outline** 3](#_Toc75288228)

[**Chapter 2: Literature Review** 4](#_Toc75288229)

[**Chapter 3: Methodology** 7](#_Toc75288230)

[**3.1 Introduction** 7](#_Toc75288231)

[**3.2 Multivariate Linear Regression** 7](#_Toc75288232)

[**3.2.1 Multivariate Linear Regression (Batch Learning Setting)** 7](#_Toc75288233)

[**3.2.2 Incremental Learning Setting** 8](#_Toc75288234)

[**3.3 Experimental Methodology** 9](#_Toc75288235)

[**3.3.1 Data Acquisition** 10](#_Toc75288236)

[**3.3.2 Data Processing** 10](#_Toc75288237)

[**3.3.3 Feature Extraction** 10](#_Toc75288238)

[**3.3.4 Incremental Learning** 10](#_Toc75288239)

[**Chapter 4: Experimental Results** 11](#_Toc75288240)

[**4.1 Introduction** 11](#_Toc75288241)

[**4.2 Design and Implementation** 11](#_Toc75288242)

[**4.3 Result Analysis** 11](#_Toc75288243)

[**Chapter 5: Conclusion and Future Work** 17](#_Toc75288244)

[**References** 18](#_Toc75288245)

[**Appendix (Source Code)** 23](#_Toc75288246)

**List of Figures**

[**Figure 1: Computational process of incremental learning** 9](file:///E:\aiub\12th%20sem\thesis\thesis-book\thesis-book-v4\thesis-book-v4.docx#_Toc75288329)

[**Figure 2: Experimental result for Dhaka Bank Ltd.** 12](#_Toc75288330)

[**Figure 3: Experimental result for BRAC Bank Ltd.** 13](#_Toc75288331)

[**Figure 4: Experimental result for BEXIMCO Pharmaceuticals Ltd.** 13](#_Toc75288332)

[**Figure 5: Experimental result for ACME Laboratories Ltd.** 14](#_Toc75288333)

[**Figure 6: Experimental result for Aramit Cement Ltd.** 15](#_Toc75288334)

[**Figure 7: Experimental result for Confidence Cement Ltd.** 15](#_Toc75288335)

# **Abstract**

The endeavour of predicting stock prices using different mathematical and technological methods and tools is not new. But the recent advancements and curiosity regarding big data and machine learning has added a new dimension to it. In this research study we investigated the feasibility and performance of multivariate linear regression method in the prediction of stock prices. Here, the multivariate linear regression was used on the basis of incremental machine learning setting.

The study conducted an experiment to predict the closing price of stocks of six different organizations enlisted in Dhaka Stock Exchange (DSE). Three years of historical stock market data (2017-2019) of these organizations have been used. Here, the Multivariate Linear Regression, Squared Loss Function, and Stochastic Gradient Descent (SGD) algorithm have been used as predictor, loss function and optimizer respectively. The model incrementally learned from the data of several stock related attributes and predicted the closing price of next day. The performance of prediction was then analysed and assessed on the basis of rolling Mean Absolute Error (MAE) metric. The rolling MAE scores for six organizations were 0.074, 0.339, 3.021, 1.284, 0.270 and 2.602 which is quite promising.

# **Chapter 1: Introduction**

## **1.1 Introduction**

Stock market is one of the most stochastic financial sectors of today’s world. The economic condition of a country is largely impacted by the movement of its stock market. Although the Efficient Market Hypothesis (EMH) indicates the impossibility of consistently predicting the movements of the market, the practice of leveraging the power of machine learning techniques in stock trading is rising day by day [1]. Along with fundamental and technical analysis which are being used traditionally, several machine learning algorithms seem to be effective up to some extent in some cases [2]. The rapid improvement of the capability of computing devices has simplified the implementation of these highly data intensive algorithms. Besides, the real-time predictive analytics is getting much more attention in recent times especially in highly dynamic economic and financial sectors. Since the stock market data can be considered as streaming data, stock trading can also be benefited by using real-time predictive analytics [3]. In terms of machine learning for streaming data the approach of incremental learning is often more helpful than its batch learning counterpart [4, 5]. There are several machine learning algorithms which are suitable to use based on the setting of incremental learning. One such algorithm is Linear Regression which is very effective in the prediction of time-series data like those of stock market [6].

## **1.2 Problem Statement**

While the batch learning approach of machine learning has its own set of advantages, the necessity of incremental learning is increasing day by day. In today’s stock market of high frequency trading this necessity is significantly relevant. An automated stock prediction system based on incremental learning mechanism could be of great use in terms of getting the market insight and making decisions instantaneously.

## **1.3 Scope of the Research**

This research study is conducted using a multivariate linear regression model. The model is designed and executed based on incremental learning setting. One of the aspects of incremental learning for streaming data is the ability of learning and predicting instantaneously which demands a model that should be computationally less expensive. A simple but effective machine learning algorithm like linear regression is a suitable candidate in this regard [7]. The study used stock data of six organizations (Dhaka Bank Ltd., BRAC Bank Ltd., BEXIMCO Pharmaceuticals Ltd., The ACME Laboratories Ltd., Aramit Cement Ltd., and Confidence Cement Ltd.) enlisted in Dhaka Stock Exchange (DSE) [8-14]. Among these there are two private banks, two pharmaceutical companies and two cement industries. Historical stock data of three consecutive years (2017, 2018, and 2019) of these organizations have been used.

## **1.4 Objective of the Research**

The objective of this research is to come up with an incremental learning based model which could be used to predict the closing price of stocks with an acceptable degree of accuracy. The model is expected to have the ability of being used in the production environment.

## **1.5 Significance of the Research**

The nature of the stock market is highly non-stationary [15]. Therefore, there is less or no chance of a system to be developed that can impose certainty on this randomness. Notwithstanding, an AI-powered automated system capable of predicting the closing price of stocks within an acceptable range of deviations can be considered valuable in terms of getting market insights and decision making [16]. Such a system should have significant demands among stock traders and investors [17]. The outcome of this research has a potential of the development of an artificially intelligent automated stock price prediction system which could provide services to its possible clients and customers.

## **1.6 Research Outline**

This research study is organized in multiple chapters, each having multiple sections. Chapter 1 introduces the overall concepts of the research, Chapter 2 contains a glimpse and insights of the resources which have been studied for this research, Chapter 3 focuses on the methodology used to conduct the research, Chapter 4 contains the analysis of the experimental results and finally Chapter 5 concludes the overall research findings and proposes some related future works based on the findings.

Any formal attempt of predicting the movement of stock market and providing it as a service is undoubtedly a very serious matter. It involves the financial resources of people. Therefore, special attention and consciousness should be maintained to conduct a research and experiment regarding stock market. This research study tries to foster this importance and conduct the experiment accordingly.

# **Chapter 2: Literature Review**

Curiosity and attempt of predicting something is an inherent human nature. When it associates with risk and money like predicting the movement of the stock market, it becomes particularly of special interest and fascination [18]. With the advancements of computing devices and algorithmic techniques researches on predicting the stock market have been increased dramatically. The impact of various parameters on the movement of stock prices are being studied quantitatively in a more structured and rigorous way [19]. Different machine learning algorithms are being trained on these parameters, including some parameters regarding public sentiment, and their performance of prediction are being evaluated from different points of view.

With the unprecedented increase in the amount of data the technologies of managing and utilizing those are being improved rapidly [20]. Business organizations are realizing the necessity of using the insight of these tremendous amount of data to make more effective data-driven decisions [21]. Many of these data come in the form of streaming data from different sources. Often it seems helpful to get insight and predict trends based on these data instantaneously. The necessity of incremental learning is realized in this context. Incremental learning has several benefits over batch learning. Unlike batch learning incremental learning doesn’t forget previous knowledge which makes this approach of learning even more natural [22]. Another advantage is- in the case of incremental learning data can be loaded into the memory of the computer successively instead of all at once [23]. Therefore, if the dataset is too large to fit into the memory, incremental learning can be the option of choice [24]. Stock market data are dynamic and have streaming nature. Very often decisions are required to be made on the fly in a high frequency trading environment. This is a suitable scenario where the techniques of incremental learning can be implemented.

There are many types and aspects of machine learning algorithms. Some algorithms are designed for supervised learning problems, some are for unsupervised learning problems [25, 26]. There are also other paradigms such as semi-supervised learning and reinforcement learning [27-31].

Predicting stock prices is a regression problem which falls under the supervised learning category. Many algorithms have been devised and tested for solving regression problem. Algorithms like Linear Regression and Support vector machine are linear in nature [32]. On the other hand ensemble methods like Random Forest, Bagging, Boosting etc. are composed of multiple models [33]. Implementation of neural networks has gained much popularity recent years [34-36]. The power of neural networks comes with the cost of powerful computing devices and time. This computational expenses sometimes make neural networks unsuitable for many applications.

S. Banerjee, N. Dabeeru and R. Lavanya tried to get a suitable regression model among simple and multiple linear regression to predict the stock prices in their paper [37]. Where multiple linear regression model seems to do better. Besides, the usability of Principal Component Analysis (PCA) and Support Vector Machine (SVM) in cooperation with multiple linear regression is also demonstrated. The paper avoids the intangible parameters like human sentiment, social media influence, reputation of the company etc. which might have influence on the stock market.

The research of Md. F. Hossain, P. Chakraborty and S. Islam is aimed to investigate the applicability of SVM in the prediction of stock prices through experiments [38]. This paper used data of Dhaka Stock Exchange (DSE) [14]. The performance of SVM was compared to some other traditional methods like Single Exponential Smoothing (SES), Double Exponential Smoothing (DES) and Auto Regressive Integrated Moving Average (ARIMA). Among these the SVM based model seems to do better.

In their study N. Hasan and R. I. Rasel proposed for using Artificial Neural Network (ANN) which is basically a three-layer perceptron model (a feed-forward neural network) in association with windowing operator [39]. Here, the sigmoid function was used as the activation function and the Sliding Window Validation was used as a validation technique. The evaluation metrics used were RMSE and MAPE.

The objective of the paper of Mustain B. Rubel et al. is to explore an efficient method for predicting the closing price of different stocks of Dhaka Stock Exchange (DSE) [40]. They applied Artificial Neural Network (ANN) and Adaptive Neuro-Fuzzy Inference System (ANFIS) on the historical stock data of several companies. The research found ANFIS as a significantly effective technique for predicting closing prices.

The paper of David M. Q. Nelson et al. investigates the applicability of Long Short-Term Memory (LSTM) networks, which is a variant of Recurrent Neural Network (RNN), to anticipate future trends of the prices of stocks using historical data [41]. They also considered some indicators of technical analysis. The model was assessed by several metrics. The performance was compared with that of some other machine learning algorithms to test its capability. The results obtained were quite satisfactory. It has 55.9% accuracy on an average in terms of predicting the movement of the prices of stocks in the imminent future.

G. Montana and F. Parrella proposed an algorithm based on Support Vector Regression (SVR) to mitigate the drawbacks of batch learning and incorporate incremental learning ability for streaming data [42]. They tried to solve the problem in the perspective of algorithmic trading.

There are several techniques for solving regression problem and new techniques are being developed and improved regularly. While traditional techniques like simple and multiple linear regression, Auto Regressive Integrated Moving Average (ARIMA), Support Vector Machine (SVM) are common in use, more sophisticated techniques like Random Forest, Bagging, Boosting and Neural Networks are gaining attention as well. Computational expenses can make many of the sophisticated techniques unsuitable for some applications. In those cases, traditional techniques might be considered more suitable. Another aspect is- some algorithms are inherently capable of working based on an incremental learning setting. Some need to be modified or redesigned to accomplish the goal.

It is important to keep in mind that there is no algorithm which is perfect for all cases no matter how much sophisticated it is. Every algorithm or technique has its own set of advantages and disadvantages. Their applicability is mainly context dependent. Therefore, the nature of the context or problem should be considered before choosing an algorithm.

# **Chapter 3: Methodology**

## **3.1 Introduction**

This research study is done with a particular focus on implementation. Required theoretical studies have been done before conducting experiments. A methodology was designed to specify the workflow of the experiments. The purpose of experiments was to train a machine learning model (multivariate linear regression) incrementally on the historical stock data of six organizations enlisted in Dhaka Stock Exchange (DSE) in order to predict the closing price [14]. Three years of historical stock data (2017-2019) of these organizations have been taken in consideration for training the model and prediction. The performance of prediction was then evaluated using the rolling Mean Absolute Error (MAE) metric.

## **3.2 Multivariate Linear Regression**

Multivariate linear regression is an extended version of univariate linear regression. Here, more than one independent variables are involved unlike its univariate counterpart. Therefore, all the associated methods and techniques require to be modified in order to comply with the requirements of the configuration of multiple variables.

### **3.2.1 Multivariate Linear Regression (Batch Learning Setting)**

In the case of multivariate linear regression we try to predict a target variable (dependent variable) based on multiple feature variables (independent variables) [43]. To accomplish this goal a hypothesis (which plays role as a predictor) should be constructed as a function of the feature variables. This is a linear function which can be expressed generally as-

---------------------------(ⅰ)

Here, are feature variables and is the target variable. are parameters.

The hypothesis function (ⅰ) tries to map the feature variables with the target variable based on different values of the parameters . This mapping actually provides an estimated value of . Each combination of the values of parametersprovides an estimated value of for each value of . Here the difference of the actual and estimated values of will indicate whether the set of parameter values is optimal or not. The optimal set of the values of parameters would be that one for which this difference is minimum. To check the optimality based on difference another function is used which is called Cost Function [44]. The cost function can be expressed in many ways. One representation (a modified version of the Mean Squared Error Function) is as follows-

----------------------------(ⅱ)

Here, *m* is the number of total training samples.

The cost function (ⅱ) can be plotted in a multi-dimensional space where the independent variables are and the dependent variable is the cost, . Our goal is to get the minimum value of the cost function (the minimum cost). The set of the values of the parameters associated with this minimum cost is considered optimal. For this set of the parameters the hypothesis provides the best fitted straight line which maps the features with the target in the best possible way.

In order to get the minimum value of the cost function another function is used which is called Optimization Function (or Optimizer) [45]. There are several functions which can be used as an optimizer. A frequently used optimizer is Gradient Descent which can be expressed (for the cost function, ) as shown in (ⅲ) [46]. The process is repeated until convergence.

-----------------------------(ⅲ)

Here, is the learning rate that determines how fast the intended minimum of the cost function would be approached. Special care should be taken in determining the value of , because a too small value of can make the gradient descent process significantly slow. On the other hand, a too large value of can cause the gradient descent to overshoot the minimum. Consequently it may fail to converge or even diverge.

In an ideal case, the gradient descent process is expected to converge to the minimum value of the cost function for which the associated set of parameters would be considered optimal. The hypothesis function will then provide the best-fitted line (best prediction) based on these set of parameters.

### **3.2.2 Incremental Learning Setting**

Incremental learning deals with the concept drift of the relevant parameters of a system [47]. In contrast to the batch learning, the focus of incremental learning is mainly on the adjustment of this concept drift. Therefore, the incremental version of the multivariate linear regression differs from its batch counterpart basically in the formulation of the cost function and optimizer. While the batch learning takes all the training samples at once, incremental learning takes only one sample at a time. Hence, the formulation of the cost function and optimizer is done based on only one sample instead of the whole training samples in an incremental mode [46].

Thus, the incremental version of the cost function becomes as shown in (ⅳ).

-----------------------------------(ⅳ)

And the incremental version of the gradient descent method becomes as shown in (ⅴ). The process is repeated until convergence. It is noticeable that the gradient descent method of (ⅴ) resembles that of (ⅲ), but unlike the method of (ⅲ) it optimizes the parameters for only one sample at a time.

-----------------------------(ⅴ)

Another useful aspect regarding incremental learning is the way of its implementation. The fact is- there is no hard and fast rule on how to work with an incremental learning algorithm. But a typical workflow can be expressed as the diagram of Figure 1 [48].

Start

Take one sample

Predict the target

Evaluate the performance of the prediction

If there are more samples

Check for structural break or concept drift in the model

Fit the model to the sample

End

Figure 1: Computational process of incremental learning

## **3.3 Experimental Methodology**

A methodology determines the structure and workflow of an experiment. The methodology considered in this study consists of four steps. These are- Data Acquisition, Data Processing, Feature Extraction, and Incremental Learning. Each step involves a bunch of tasks to prepare the input for the next step and ensure the overall smooth execution. Iteration through these steps is also possible if required. In this study the tasks associated with each step have been done through manual coding but in an ideal production environment the automated system should execute the whole workflow in an automated manner in nearly real-time.

### **3.3.1 Data Acquisition**

This is the first experimental step. In this step data are collected in order to conduct the experiment. For many incremental learning problems nearly real-time data are available along with historical data. For example, during trading period stock market data can be collected as nearly real-time data.

### **3.3.2 Data Processing**

Typically data acquired from the original sources are unprocessed. In this step some data processing tasks are done in order to convert those into the intended format required in the following steps. Some data processing tasks include cleaning missing data and outliers, transforming and reducing data etc.

### **3.3.3 Feature Extraction**

Not all features of data which come from the original sources are important for the experiment. In this step the unnecessary features are discarded. Sometimes new features might need to be derived from existing features. In this case different techniques of feature engineering can be helpful.

### **3.3.4 Incremental Learning**

This is the actual machine learning step. In this step a machine learning model is used as a predictor. This model needs to be accompanied by a cost function and an optimizer [44, 45]. These methods can be put into action based on the workflow of incremental learning shown in Figure 1.

The methodology used in incremental learning setting is pretty much straightforward like those used in typical machine learning experiments. The main difference of the workflow of incremental learning from that of batch learning is in the mechanism of model fitting. Unlike batch learning an incremental learning model is fitted to only one sample at a time based on the detected concept drift.

# **Chapter 4: Experimental Results**

## **4.1 Introduction**

The experimental setup of a problem is mainly determined by the nature of the problem. Predicting stock prices is a regression problem on time-series data which requires some special care so that the time-series nature of data remains conserved at every phase of the experiment. Keeping this in mind the experiment was designed carefully. Another important aspect regarding any experiment is bias which can be introduced consciously or subconsciously. Bias can skew the result of an experiment towards a certain direction. So, special attention should be given to address this possible issue.

## **4.2 Design and Implementation**

In this research study experiments were conducted using Python programming language. To perform Exploratory Data Analysis (EDA) and machine learning related tasks several Python libraries were used [49]. These are Pandas, Numpy, Matplotlib, and River (library for incremental learning) [50-53]. To conduct the experiment three years of historical data (2017-2019) of selected organizations was collected [14]. The data was then processed in order to remove noises. After that unnecessary features were removed from the required data. Several strategies of normalization and transformation were also implemented. Exploratory data analysis was done to discover the nature of data. This helped to determine some techniques and hyper-parameters later on. After that a multivariate linear regression model was used as a hypothesis (predictor) function. The incremental version of the Squared Loss function and Stochastic Gradient Descent (SGD) algorithm were used as the loss function and optimizer respectively [44, 45].

Here, nine feature variables (regarding the trading of each day) were used to predict the target variable (closing price). These features were date, last traded price, highest price, lowest price, opening price, yesterday’s closing price, total trade, value (in million), and volume. The closing price of next day of a particular stock were predicted and the performance of prediction was evaluated in terms of rolling Mean Absolute Error (MAE) [54]. The rolling window size was 12. That means the average score of MAE of previous 12 days was considered to update the metric for each sample.

## **4.3 Result Analysis**

Experiments have been done multiple times to get an optimal set of hyper-parameters so that the results obtained are significantly reliable. Besides, special care was taken so that no known bias could be introduced. The outcomes found as a result of the experiments are given in Figures 2, 3, 4, 5, 6 and 7. Each of the figure is composed of two parts- upper and lower. The upper part is the visualization of closing price versus date graph for both of the actual and predicted prices. It also contains the latest score of rolling Mean Absolute Error (MAE). On the other hand, the lower part is the graph of MAE with respect to date.

For Dhaka Bank Ltd. (Figure 2) the latest MAE is 0.074. The graph shows that the model could predict the actual value with greater accuracy from slightly before the year 2018.

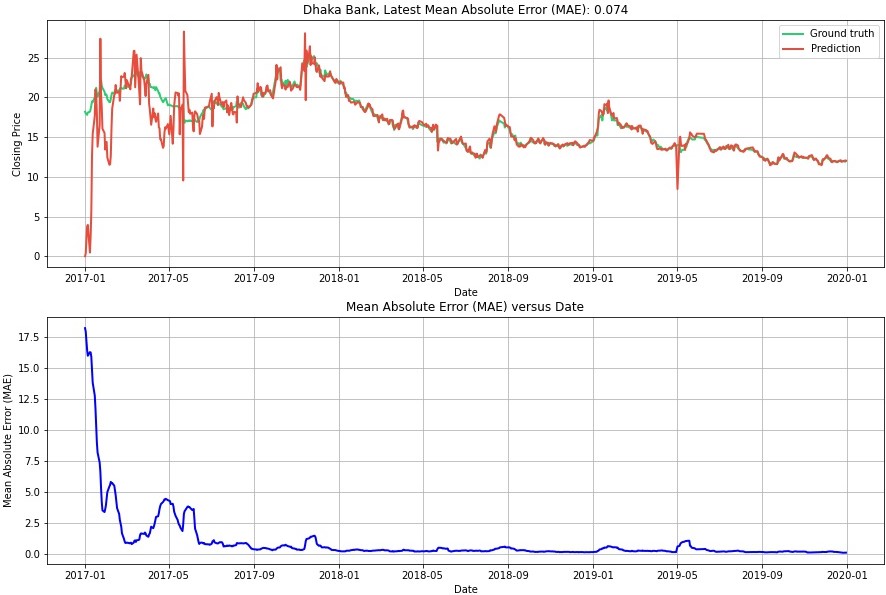


Figure 2: Experimental result for Dhaka Bank Ltd.

The latest MAE for BRAC Bank Ltd. is 0.339 (Figure 3). The model seems to catch the trend of the actual values nearly from the mid of 2017.

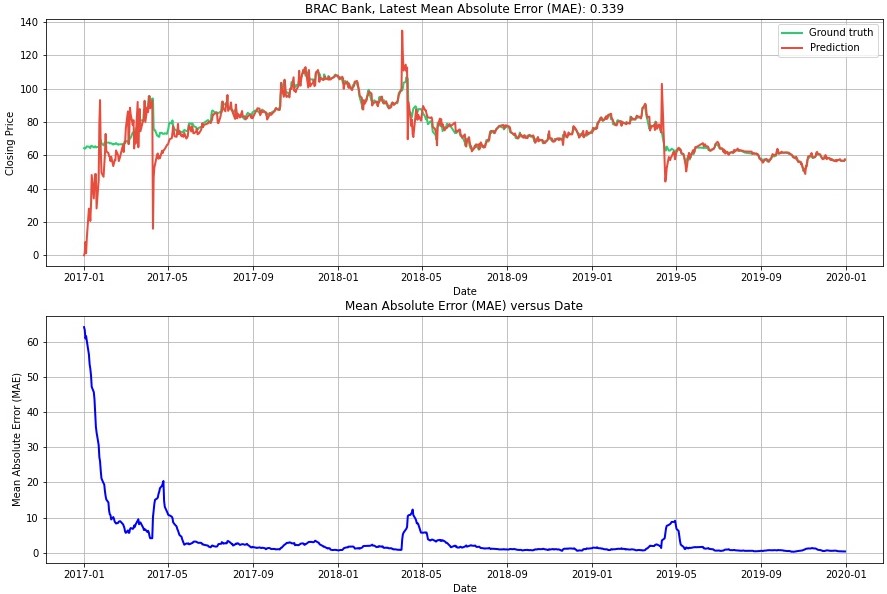


Figure 3: Experimental result for BRAC Bank Ltd.

Figure 4 shows that the latest MAE for BEXIMCO Pharmaceuticals Ltd. is 3.021. The model seems to provide several abnormal predictions, though the overall predictive performance is quite good.

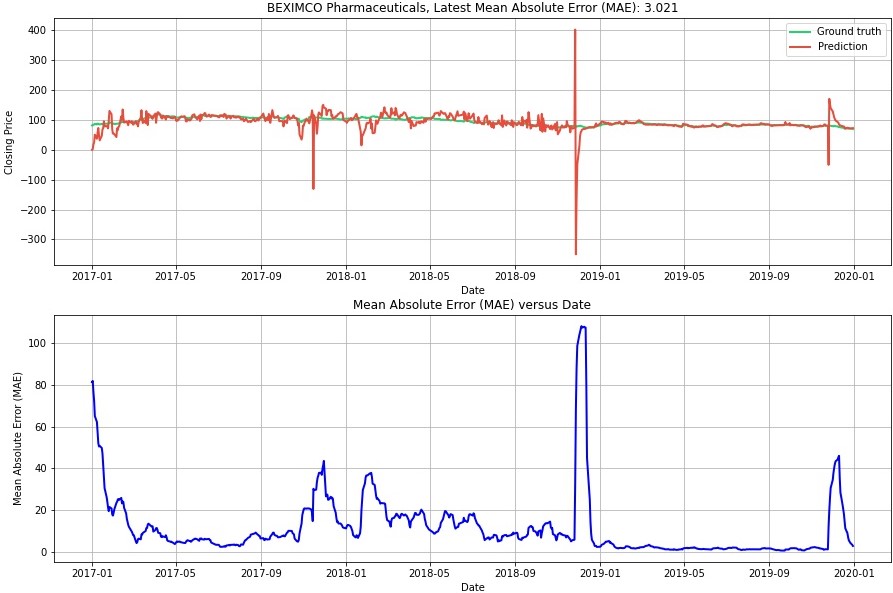


Figure 4: Experimental result for BEXIMCO Pharmaceuticals Ltd.

Figure 5 shows that the latest MAE of The ACME Laboratories Ltd. is 1.284. On the whole the model seems to capture the trend of the actual values quite well even after having several unusual predictions.



Figure 5: Experimental result for ACME Laboratories Ltd.

The latest MAE of Aramit Cement Ltd. is 0.270 (Figure 6). Even here, there are several abnormal predictions. But the overall performance is pretty good.

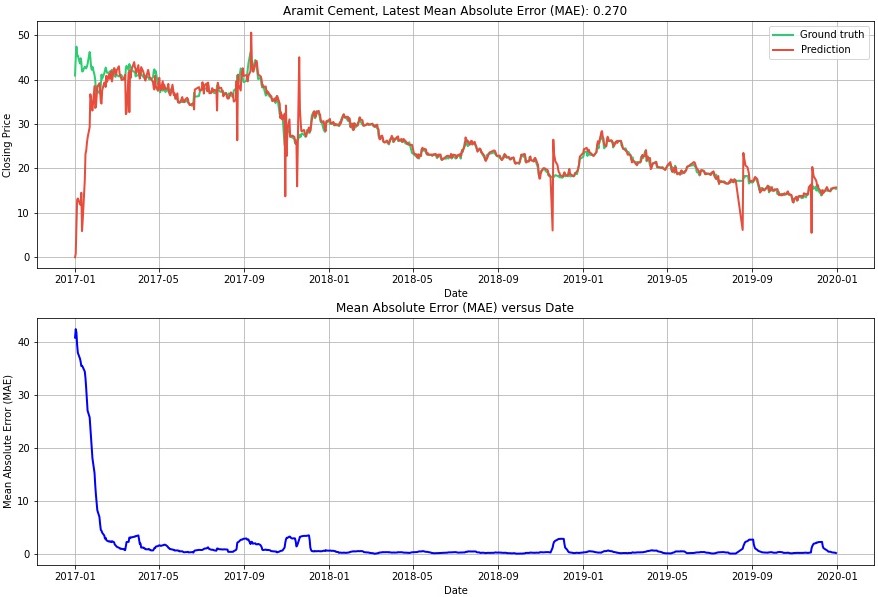


Figure 6: Experimental result for Aramit Cement Ltd.

For Confidence Cement Ltd. the latest MAE is 2.602 (Figure 7). The model seems to get confused sometimes in predicting the actual values. But its overall capability of prediction is also good.

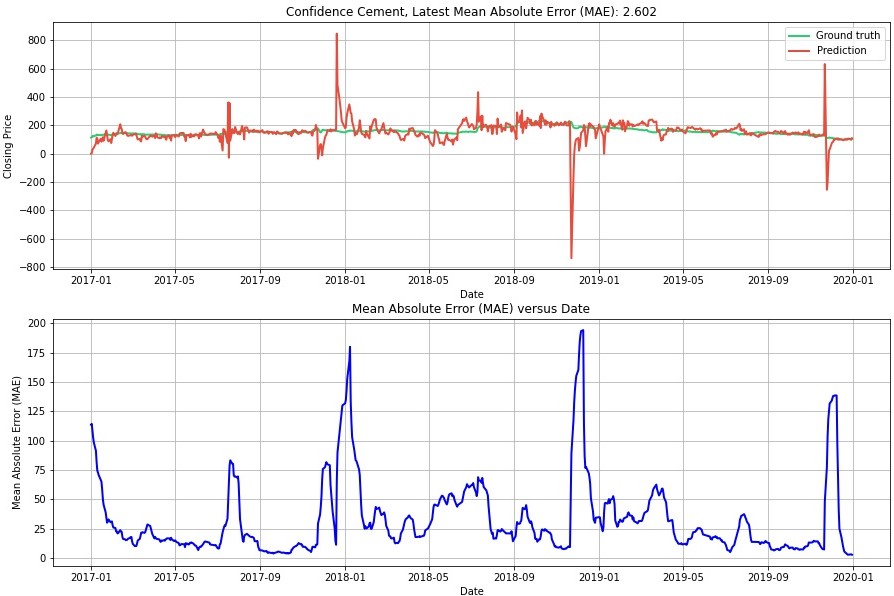


Figure 7: Experimental result for Confidence Cement Ltd.

The experimental results (Figures 2, 3, 4, 5, 6, and 7) reflect the fluctuating nature of stock market, which is quite natural. In some points of time, the actual prices for each of the organizations changed abnormally. This caused the model to be slightly confused. The abnormal spikes reflect this fact. But the model seems to update itself and comply with the latest trend very rapidly. This is the strength of the model. Besides, the changes of MAE with respect to date shown in the figures are also satisfactory. Therefore, the performance of the model seems quite promising from an overall consideration.

# **Chapter 5: Conclusion and Future Work**

Predicting the movement of stock market is not easy. There are many hidden parameters which play role in changing the trend of the market. The impact of intangible parameters which are related to national and international politics and socio-economic aspects are very hard to study in a quantitative manner. Notwithstanding, the attempt of predicting the market can provide some valuable insights to make effective data-driven decisions. This research study investigated the applicability of Multivariate Linear Regression model for predicting stock prices on an incremental learning setting. The MAE scores found for six different organizations are 0.074, 0.339, 3.021, 1.284, 0.270 and 2.602 which is quite promising. Besides, the workflow followed here has the potential of being used in the development of an incremental learning based automated stock prediction system which would be able to provide services to possible clients and customers.

The outcome of the experiments seems to satisfy the objective of the research largely. Still there are many ways of improving it. This study used data of Dhaka Stock Exchange (DSE) only [14]. Few other stock markets can be considered to test the model even further. At the same time, the incorporation of sentiment analysis into this endeavour can add a new dimension and provide a more robust model. Besides, trying some other machine learning algorithms or even neural networks as an incremental learning technique can also be helpful to come up with a more powerful model. A research study like predicting the movement of stock market is always fascinating. Throughout this research many aspects of stock market and their association in predicting the stock prices have been studied. The study found Multivariate Linear Regression with incremental learning capability as a significantly promising model. These findings can further be improved and thus hopefully a more reliable model could be found to serve the traders and investors in decision making.

# **References**

[1] Bruno M. Henrique, Vinicius A. Sobreiro, H. Kimura, “Stock price prediction using support vector regression on daily and up to the minute prices,” *The Journal of Finance and Data Science 4(3)*, April 2018, DOI: 10.1016/j.jfds.2018.04.003.

[2] Z. Iqbal et al., “Efficient machine learning techniques for stock market prediction,” *International Journal of Engineering Research and Applications*, ISSN: 2248-9622, Vol. 3, Issue 6, Nov-Dec 2013, pp.855-867.

[3] S. Nann, J. Krauss, D. Schoder, “Predictive Analytics on Public Data – The Case of Stock Markets,” *Proceedings of the 21st European Conference on Information Systems*, June 2013.

[4] H. He et al., “Incremental learning from stream data,” *IEEE Transactions on Neural Networks 22(12):1901-14*, October 2011, DOI: 10.1109/TNN.2011.2171713.

[5] Heitor M. Gomes et al., “Machine learning for streaming data: state of the art, challenges, and opportunities,” *ACM SIGKDD Explorations Newsletter 21(2):6-22*, November 2019, DOI: 10.1145/3373464.3373470.

[6] R. Seethalakshmi, “Analysis of stock market predictor variables using Linear Regression,” *International Journal of Pure and Applied Mathematics 119(15):369-377*, January 2018.

[7] Agus Q. Munir, R. Wardoyo, “Comparison Analysis of Time Series Data Algorithm Complexity for Forecasting of Dengue Fever Occurrences,” *International Journal of Advanced Computer Research 6(7)*, October 2015.

[8] “Dhaka Bank Limited,” Accessed on: March 3, 2021. [Online]. Available: <https://dhakabankltd.com/>

[9] “BRAC Bank Limited,” Accessed on: March 3, 2021. [Online]. Available: <https://www.bracbank.com/en/>

[10] “BEXIMCO Pharmaceuticals Limited,” Accessed on: May 25, 2021. [Online]. Available: <https://www.beximcopharma.com/>

[11] “The ACME Laboratories Limited,” Accessed on: May 25, 2021. [Online]. Available: <https://www.acmeglobal.com/>

[12] “Aramit Cement Limited,” Accessed on: May 25, 2021. [Online]. Available: <https://aramitcement.com/>

[13] “Confidence Cement Limited,” Accessed on: May 25, 2021. [Online]. Available: <https://www.confidencegroup.com.bd/confidence_cement.php>

[14] “Dhaka Stock Exchange Limited,” Accessed on: Feb. 17, 2021. [Online]. Available: <https://www.dsebd.org/>

[15] V. Murthy, K. Washer, John R Wingender, “Are Stock Prices in the US Nonstationary? Evidence from Contemporary Unit Root Tests,” *Applied Financial Economics 21(22):1703-1709*, November 2011, DOI: 10.1080/09603107.2011.591731.

[16] Luke K. Rose, T. Gartner, “Automated Stock Market Trading Using Machine Learning,” April 2018, DOI: 10.13140/RG.2.2.19887.59045.

[17] “*HOW AI TRADING TECHNOLOGY IS MAKING STOCK MARKET INVESTORS SMARTER*,” March 16, 2019. Accessed on: April 17, 2021. [Online]. Available: <https://builtin.com/artificial-intelligence/ai-trading-stock-market-tech>

[18] M. Usmani et al., “Stock market prediction using machine learning techniques,” *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)*, August 2016, DOI: 10.1109/ICCOINS.2016.7783235.

[19] M. Faruquee, S. Haque, “Impact of fundamental factors on stock price: a case based approach on pharmaceutical companies listed with Dhaka stock exchange,” *International Journal of Business and Management Invention ISSN (Online): 2319 – 8028*, ISSN (Print): 2319 – 801X www.ijbmi.org Volume 2 Issue 9ǁ September. 2013ǁ PP.34-41.

[20] B. Purcell, “The Emergence of “Big Data” Technology and Analytics,” *Journal of Technology Research*, January 2012.

[21] H. Koscielniak, A. Puto, “BIG DATA in Decision Making Processes of Enterprises,” Procedia Computer Science 65:1052-1058, December 2015, DOI: 10.1016/j.procs.2015.09.053.

[22] “Incremental Learning,” Accessed on: March 21, 2021. [Online]. Available: <https://en.wikipedia.org/wiki/Incremental_learning>

[23] Y. Luo et al., “An appraisal of incremental learning methods,” *Entropy 22(11):1190*, October 2020, DOI: 10.3390/e22111190.

[24] V. Losing, B. Hammer, H. Wersing, “Incremental On-line Learning: A Review and Comparison of State of the Art Algorithms,” *Neurocomputing 275*, September 2017, DOI: 10.1016/j.neucom.2017.06.084.

[25] M. Alloghani et al., “A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science,” *Supervised and Unsupervised Learning for Data Science*, January 2020, DOI: 10.1007/978-3-030-22475-2\_1.

[26] V. Nasteski, “An overview of the supervised machine learning methods,” December 2017, DOI: 10.20544/HORIZONS.B.04.1.17.P05.

[27] Y C A Padmanabha Reddy et al., “Semi-supervised learning: a brief review,” *International Journal of Engineering & Technology 7(1-8):81*, February 2018, DOI: 10.14419/ijet.v7i1.8.9977.

[28] Jesper E. van Engelen, Holger H Hoos, “A survey on semi-supervised learning,” *Springer Machine Learning 109(5)*, February 2020, DOI: 10.1007/s10994-019-05855-6.

[29] Verena Heidrich-Meisner et al., “Reinforcement learning in a Nutshell,” *15th European Symposium on Artificial Neural Networks, Bruges, Belgium, April 25-27, 2007, Proceedings*, January 2007.

[30] R. Sharma, M. Prateek, A. Sinha, “Use of Reinforcement Learning as a Challenge: A Review,” *International Journal of Computer Applications 69(22):28-34*, May 2013, DOI: 10.5120/12105-8332.

[31] S. Mousavi, M. Schukat, E. Howley, “Deep Reinforcement Learning: An Overview,” *Proceedings of SAI Intelligent Systems Conference*, June 2018, DOI: 10.1007/978-3-319-56991-8\_32.

[32] A. Pradhan, “Support vector machine-A survey,” *International Journal of Emerging Technology and Advanced Engineering*, ISSN 2250-2459, Volume 2, Issue 8, August 2012.

[33] “Ensemble Learning,” Accessed on: March 21, 2021. [Online]. Available: <https://en.wikipedia.org/wiki/Ensemble_learning>

[34] E. Grossi, M. Buscema, “Introduction to artificial neural networks,” *European Journal of Gastroenterology & Hepatology 19(12):1046-54*, August 2007, DOI: 10.1097/MEG.0b013e3282f198a0.

[35] N. Malik, “Artificial Neural Networks and their Applications,” *National Conference on ‘Unearthing Technological Developments & their Transfer for Serving Masses’*, GLA ITM, Mathura, India 17-18 April 2005.

[36] Z. Zhang, “A gentle introduction to artificial neural networks,” *Annals of Translational Medicine 4(19):370-370*, October 2016, DOI: 10.21037/atm.2016.06.20.

[37] S. Banerjee, N. Dabeeru and R. Lavanya, “Stock market prediction,” *International Journal of Innovative Technology and Exploring Engineering*, August 2020.

[38] Md. F. Hossain, P. Chakraborty and S. Islam, “Predicting Daily Closing Prices of Selected Shares of Dhaka Stock Exchange (DSE) Using Support Vector Machines,” *Internet of Things and Cloud Computing* , December 2020.

[39] N. Hasan and R. I. Rasel, “Artificial Neural Network Approach for Stock Price and Trend Prediction,” *International conference on advanced information & communication technology*, May 2016.

[40] Mustain B. Rubel et al., “Predicting Closing Stock Price using Artificial Neural Network and Adaptive Neuro Fuzzy Inference System (ANFIS): The Case of the Dhaka Stock Exchange,” *International Journal of Computer Applications*, November 2015.

[41] David M. Q. Nelson et al., “Stock Market’s Price Movement Prediction with LSTM Neural Networks,” *International Joint Conference on Neural Networks (IJCNN)*, May 2017, DOI: 10.1109/IJCNN.2017.7966019.

[42] G. Montana, F. Parrella, “Learning to Trade with Incremental Support Vector Regression Experts,” *Hybrid Artificial Intelligence Systems, Third International Workshop, HAIS 2008, Burgos, Spain*, September 24-26, 2008, DOI: 10.1007/978-3-540-87656-4\_73.

[43] Andrew Ng, “Machine Learning,” Accessed on: May 24, 2021. [Online]. Available: <https://www.coursera.org/lecture/machine-learning/multiple-features-6Nj1q>

[44] “*Data Science 101: Introduction to Cost Function*,” March 4, 2021. Accessed on: April 17, 2021. [Online]. Available: <https://www.analyticsvidhya.com/blog/2021/03/data-science-101-introduction-to-cost-function/>

[45] “*Introduction to optimizers*,” May 7, 2018. Accessed on: April 24, 2021. [Online]. Available: <https://algorithmia.com/blog/introduction-to-optimizers>

[46] “*CS 2750 Machine Learning Lecture 8-9*,” Accessed on: May 19, 2021. [Online]. Available: <https://people.cs.pitt.edu/~milos/courses/cs2750-Spring2020/Lectures/Class8.pdf>

[47] “*A Gentle Introduction to Concept Drift in Machine Learning*,” Dec. 10, 2020. Accessed on: April 13, 2021. [Online]. Available: <https://machinelearningmastery.com/gentle-introduction-concept-drift-machine-learning/>

[48] “*Linear Regression Model for Incremental Learning*,” Accessed on: May 27, 2021. [Online]. Available: <https://es.mathworks.com/help/stats/incrementalregressionlinear.html>

[49] “*Exploratory Data Analysis*,” Accessed on: March 23, 2021. [Online]. Available: <https://en.wikipedia.org/wiki/Exploratory_data_analysis>

[50] “*Pandas Documentation*,” Accessed on: March 11, 2021. [Online]. Available: <https://pandas.pydata.org/docs/>

[51] “*NumPy v1.20.0*,” Accessed on: March 11, 2021. [Online]. Available: <https://numpy.org/>

[52] “*Matplotlib v3.4.2*,” Accessed on: March 13, 2021. [Online]. Available: <https://matplotlib.org/>

[53] “*River- A Python Package for Online/Streaming Machine Learning*,” Accessed on: March 13, 2021. [Online]. Available: <https://riverml.xyz/latest/>

[54] L. Hallman, “The Rolling Window Method: Precisions of Financial Forecasting,” *Degree Project in Mathematics*, Second Cycle, KTH, Stockholm, Sweden, 2017.

# **Appendix (Source Code)**

**Exploratory Data Analysis (EDA)**

**# Import libraries**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

from river import stream

import datetime

**# Import training data**

df\_2017 = pd.read\_json("./data/dse-dataset/prices\_2017.json")

df\_2018 = pd.read\_json("./data/dse-dataset/prices\_2018.json")

df\_2019 = pd.read\_json("./data/dse-dataset/prices\_2019.json")

**# Combine all the dataFrames into a single one**

df\_complete = pd.concat([df\_2017, df\_2018, df\_2019])

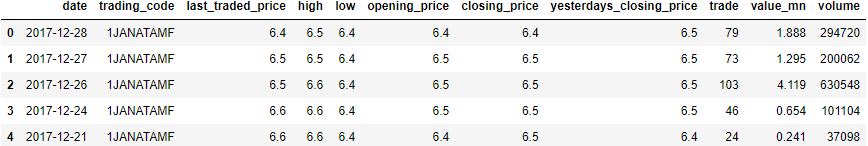
**# View the shape**

df\_complete.shape



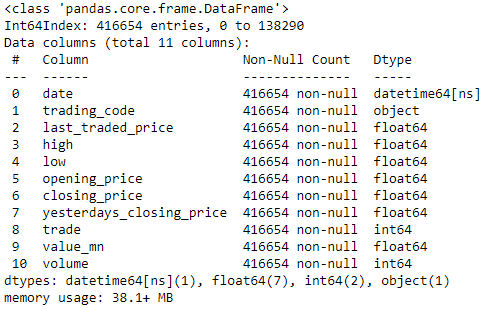
**# View first 5 instances**

df\_complete.head()



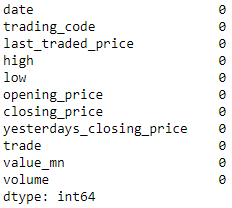
**# Get basic information about data**

df\_complete.info()



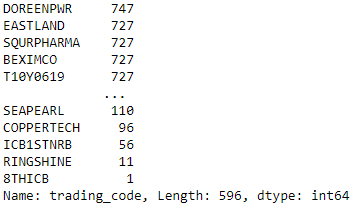
**# Check if there is any null value**

df\_complete.isna().sum()



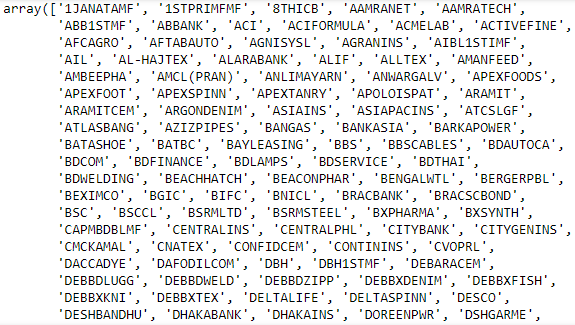
**# Get the unique companies along with the number of their samples**

df\_complete.trading\_code.value\_counts()



**# Get the list of unique companies**

df\_complete.trading\_code.unique()



**# Check if the trading code of Dhaka bank (DHAKABANK) is present**

list(df\_complete.trading\_code.unique()).count("DHAKABANK")



**# Check if the trading code of BRAC bank (BRACBANK) is present**

list(df\_complete.trading\_code.unique()).count("BRACBANK")



**# Check if the trading code of BEXIMCO Pharma (BXPHARMA) is present**

list(df\_complete.trading\_code.unique()).count("BXPHARMA")



**# Check if the trading code of ACME Lab (ACMELAB) is present**

list(df\_complete.trading\_code.unique()).count("ACMELAB")



**# Check if the trading code of Aramit Cement (ARAMITCEM) is present**

list(df\_complete.trading\_code.unique()).count("ARAMITCEM")



**# Check if the trading code of Confidence Cement (CONFIDCEM) is present**

list(df\_complete.trading\_code.unique()).count("CONFIDCEM")



**# Create distinct dataFrame for each of DHAKABANK, BRACBANK, BXPHARMA, ACMELAB, ARAMITCEM and CONFIDCEM**

df\_dhaka\_bank = df\_complete[df\_complete.trading\_code == "DHAKABANK"]

df\_brac\_bank = df\_complete[df\_complete.trading\_code == "BRACBANK"]

df\_beximco\_pharma = df\_complete[df\_complete.trading\_code == "BXPHARMA"]

df\_acme\_lab = df\_complete[df\_complete.trading\_code == "ACMELAB"]

df\_aramit\_cement = df\_complete[df\_complete.trading\_code == "ARAMITCEM"]

df\_confidence\_cement = df\_complete[df\_complete.trading\_code == "CONFIDCEM"]

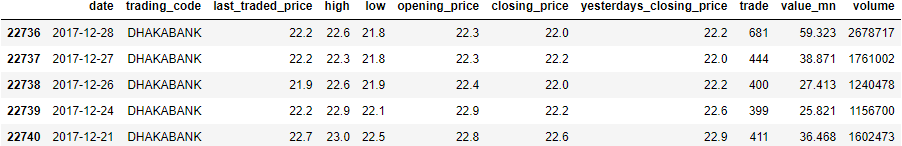
**# Get total number of samples for each bank**

len(df\_dhaka\_bank), len(df\_brac\_bank), len(df\_beximco\_pharma), len(df\_acme\_lab), len(df\_aramit\_cement),len(df\_confidence\_cement)



**# View first 5 samples of Dhaka bank**

df\_dhaka\_bank.head()



**# Discard unnecessary column**

df\_dhaka\_bank = df\_dhaka\_bank.drop("trading\_code", axis=1)

df\_brac\_bank = df\_brac\_bank.drop("trading\_code", axis=1)

df\_beximco\_pharma = df\_beximco\_pharma.drop("trading\_code", axis=1)

df\_acme\_lab = df\_acme\_lab.drop("trading\_code", axis=1)

df\_aramit\_cement = df\_aramit\_cement.drop("trading\_code", axis=1)

df\_confidence\_cement = df\_confidence\_cement.drop("trading\_code", axis=1)

**# Sort the samples by ascending order of date**

df\_dhaka\_bank.sort\_values(by=["date"], inplace=True, ascending=True)

df\_brac\_bank.sort\_values(by=["date"], inplace=True, ascending=True)

df\_beximco\_pharma.sort\_values(by=["date"], inplace=True, ascending=True)

df\_acme\_lab.sort\_values(by=["date"], inplace=True, ascending=True)

df\_aramit\_cement.sort\_values(by=["date"], inplace=True, ascending=True)

df\_confidence\_cement.sort\_values(by=["date"], inplace=True, ascending=True)

**# View first 5 samples (sorted) of Dhaka bank**

df\_dhaka\_bank.head()



**# Plot closing\_price with respect to date**

fig, (ax0,ax1,ax2,ax3,ax4,ax5) = plt.subplots(nrows=6, ncols=1, figsize=(15,31))

**# Add data for ax0**

ax0.plot(df\_dhaka\_bank.date, df\_dhaka\_bank.closing\_price, lw=2, alpha=1)

**# Customize ax0**

ax0.grid(alpha=0.95)

ax0.set(title="Dhaka Bank",

xlabel="Date",

ylabel="Closing Price")

**# Add data for ax1**

ax1.plot(df\_brac\_bank.date, df\_brac\_bank.closing\_price, lw=2, alpha=1)

**# Customize ax1**

ax1.grid(alpha=0.95)

ax1.set(title="BRAC Bank",

xlabel="Date",

ylabel="Closing Price")

**# Add data for ax2**

ax2.plot(df\_beximco\_pharma.date, df\_beximco\_pharma.closing\_price, lw=2, alpha=1)

**# Customize ax2**

ax2.grid(alpha=0.95)

ax2.set(title="BEXIMCO Pharma",

xlabel="Date",

ylabel="Closing Price")

**# Add data for ax3**

ax3.plot(df\_acme\_lab.date, df\_acme\_lab.closing\_price, lw=2, alpha=1)

**# Customize ax3**

ax3.grid(alpha=0.95)

ax3.set(title="ACME Laboratories",

xlabel="Date",

ylabel="Closing Price")

**# Add data for ax4**

ax4.plot(df\_aramit\_cement.date, df\_aramit\_cement.closing\_price, lw=2, alpha=1)

**# Customize ax4**

ax4.grid(alpha=0.95)

ax4.set(title="Aramit Cement",

xlabel="Date",

ylabel="Closing Price")

**# Add data for ax5**

ax5.plot(df\_confidence\_cement.date, df\_confidence\_cement.closing\_price, lw=2, alpha=1)

**# Customize ax5**

ax5.grid(alpha=0.95)

ax5.set(title="Confidence Cement",

xlabel="Date",

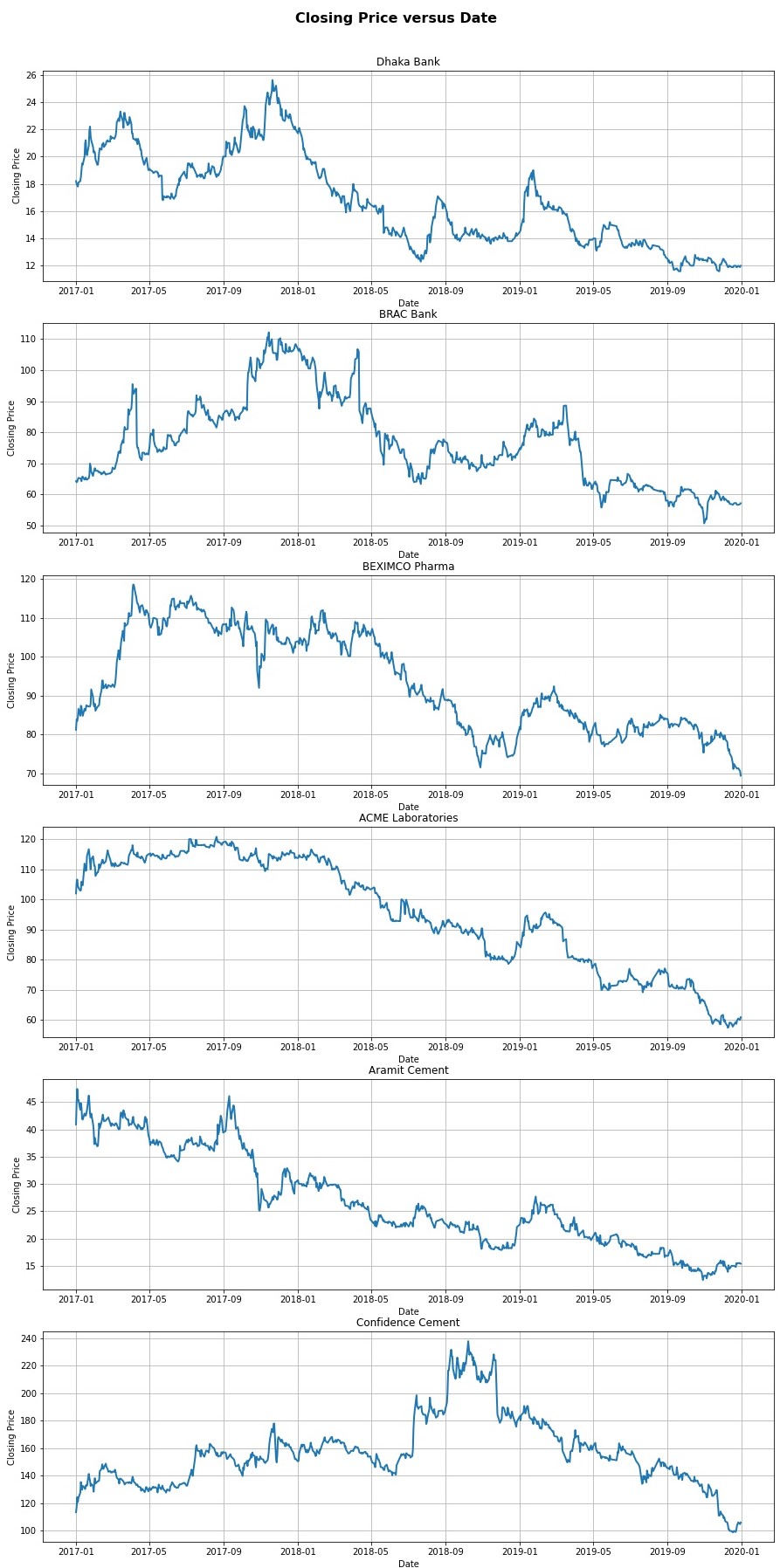
ylabel="Closing Price")

**# Title the figure**

fig.suptitle("Closing Price versus Date", fontsize=16, fontweight='bold', y=0.91)

**# Save the figure**

plt.savefig("./figures/closing-price-versus-date.jpg")



**# Export data as a csv file**

df\_dhaka\_bank.to\_csv("./data/dse-final-data/dhaka\_bank\_data.csv", index=False)

df\_brac\_bank.to\_csv("./data/dse-final-data/brac\_bank\_data.csv", index=False)

df\_beximco\_pharma.to\_csv("./data/dse-final-data/beximco\_pharma\_data.csv", index=False)

df\_acme\_lab.to\_csv("./data/dse-final-data/acme\_lab\_data.csv", index=False)

df\_aramit\_cement.to\_csv("./data/dse-final-data/aramit\_cement\_data.csv", index=False)

df\_confidence\_cement.to\_csv("./data/dse-final-data/confidence\_cement\_data.csv", index=False)

**Let's import data as streaming data, convert string format date to date format date and investigate the data types of some other variables (for testing purpose).**

for x, y in stream.iter\_csv("./data/dse-final-data/dhaka\_bank\_data.csv", target='closing\_price'):

date\_list = x["date"].split("-")

for index, element in enumerate(date\_list):

if index == 0:

year = int(element)

elif index == 1:

month = int(element)

elif index == 2:

day = int(element)

print(year, month, day)

print(type(year), type(month), type(day))

x["date"] = datetime.date(year, month, day)

x["last\_traded\_price"] = float(x["last\_traded\_price"])

x["high"] = float(x["high"])

x["low"] = float(x["low"])

x["opening\_price"] = float(x["opening\_price"])

x["yesterdays\_closing\_price"] = float(x["yesterdays\_closing\_price"])

x["trade"] = int(x["trade"])

x["value\_mn"] = float(x["value\_mn"])

x["volume"] = int(x["volume"])

print(x, y)

print("Type of x: ", type(x))

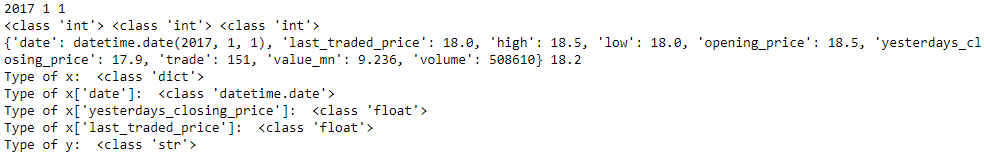
print("Type of x['date']: ", type(x["date"]))

print("Type of x['yesterdays\_closing\_price']: ", type(x["yesterdays\_closing\_price"]))

print("Type of x['last\_traded\_price']: ", type(x["last\_traded\_price"]))

print("Type of y: ", type(y))

break



**Now we will define a function for converting the date to ordinal date. Also we will pre-process, normalize and instantiate the model using the pipeline.**

**# Import libraries**

from river import compose

from river import linear\_model

from river import preprocessing

from river import optim

**#Initialize the SGD optimizer**

optimizer = optim.SGD()

**#Initialize the Squared Loss (L2 Loss) function**

loss = optim.losses.Squared()

**# Create function for converting the format of date**

def get\_ordinal\_date(x):

return {'ordinal\_date': x['date'].toordinal(), "last\_traded\_price": x["last\_traded\_price"], "high": x["high"], "low": x["low"], "opening\_price": x["opening\_price"], "yesterdays\_closing\_price": x["yesterdays\_closing\_price"], "trade": x["trade"], "value\_mn": x["value\_mn"], "volume": x["volume"]}

**# Instantiate models**

model\_1 = compose.Pipeline(

('ordinal\_date', compose.FuncTransformer(get\_ordinal\_date)),

('scale', preprocessing.StandardScaler()),

('lin\_reg', linear\_model.LinearRegression(optimizer, loss))

)

model\_2 = compose.Pipeline(

('ordinal\_date', compose.FuncTransformer(get\_ordinal\_date)),

('scale', preprocessing.StandardScaler()),

('lin\_reg', linear\_model.LinearRegression(optimizer, loss))

)

model\_3 = compose.Pipeline(

('ordinal\_date', compose.FuncTransformer(get\_ordinal\_date)),

('scale', preprocessing.StandardScaler()),

('lin\_reg', linear\_model.LinearRegression(optimizer, loss))

)

model\_4 = compose.Pipeline(

('ordinal\_date', compose.FuncTransformer(get\_ordinal\_date)),

('scale', preprocessing.StandardScaler()),

('lin\_reg', linear\_model.LinearRegression(optimizer, loss))

)

model\_5 = compose.Pipeline(

('ordinal\_date', compose.FuncTransformer(get\_ordinal\_date)),

('scale', preprocessing.StandardScaler()),

('lin\_reg', linear\_model.LinearRegression(optimizer, loss))

)

model\_6 = compose.Pipeline(

('ordinal\_date', compose.FuncTransformer(get\_ordinal\_date)),

('scale', preprocessing.StandardScaler()),

('lin\_reg', linear\_model.LinearRegression(optimizer, loss))

)

**In order to evaluate the model we'll write down a function. This function will take each sample one by one. It will perform a prior prediction and then update the model and error metric. Finally the predicted values will be plotted along with the actual values.**

**# Import libraries**

from river import metrics

from river import stream

import matplotlib.pyplot as plt

**# Create evaluation function**

def evaluate\_model(model, company\_name, data\_path):

metric = metrics.Rolling(metrics.MAE(), 12)

dates = []

y\_trues = []

y\_preds = []

metric\_list = []

for x, y in stream.iter\_csv(data\_path, target='closing\_price'):

**#Convert string format date to actual date**

date\_list = x["date"].split("-")

for index, element in enumerate(date\_list):

if index == 0:

year = int(element)

elif index == 1:

month = int(element)

elif index == 2:

day = int(element)

x["date"] = datetime.date(year, month, day)

x["last\_traded\_price"] = float(x["last\_traded\_price"])

x["high"] = float(x["high"])

x["low"] = float(x["low"])

x["opening\_price"] = float(x["opening\_price"])

x["yesterdays\_closing\_price"] = float(x["yesterdays\_closing\_price"])

x["trade"] = int(x["trade"])

x["value\_mn"] = float(x["value\_mn"])

x["volume"] = int(x["volume"])

y = float(y)

**# perform the prior prediction and update the model**

y\_pred = model.predict\_one(x)

model.learn\_one(x, y)

**# Update the error metric**

metric.update(y, y\_pred)

**# Store the date, true value, predicted value and metric value**

dates.append(x['date'])

y\_trues.append(y)

y\_preds.append(y\_pred)

metric\_list.append(metric.get())

**# Plot the results**

fig, (ax0, ax1) = plt.subplots(nrows=2, ncols=1, figsize=(15, 10))

ax0.grid(alpha=0.95)

ax0.plot(dates, y\_trues, lw=2, color='#2ecc71', alpha=1, label='Ground truth')

ax0.plot(dates, y\_preds, lw=2, color='#e74c3c', alpha=1, label='Prediction')

ax0.legend()

ax0.set(title="{}, Latest Mean Absolute Error (MAE): {:.3f}".format(company\_name, metric.get()),

xlabel="Date",

ylabel="Closing Price")

ax1.grid(alpha=0.95)

ax1.plot(dates, metric\_list, lw=2, color='blue', alpha=1)

ax1.set(title="Mean Absolute Error (MAE) versus Date",

xlabel="Date",

ylabel="Mean Absolute Error (MAE)")

plt.savefig("./figures/{}.jpg".format(company\_name))

**Now we’ll specify data path.**

**# Specifying data path**

dhaka\_bank\_data\_path = "./data/dse-final-data/dhaka\_bank\_data.csv"

brac\_bank\_data\_path = "./data/dse-final-data/brac\_bank\_data.csv"

beximco\_pharma\_data\_path = "./data/dse-final-data/beximco\_pharma\_data.csv"

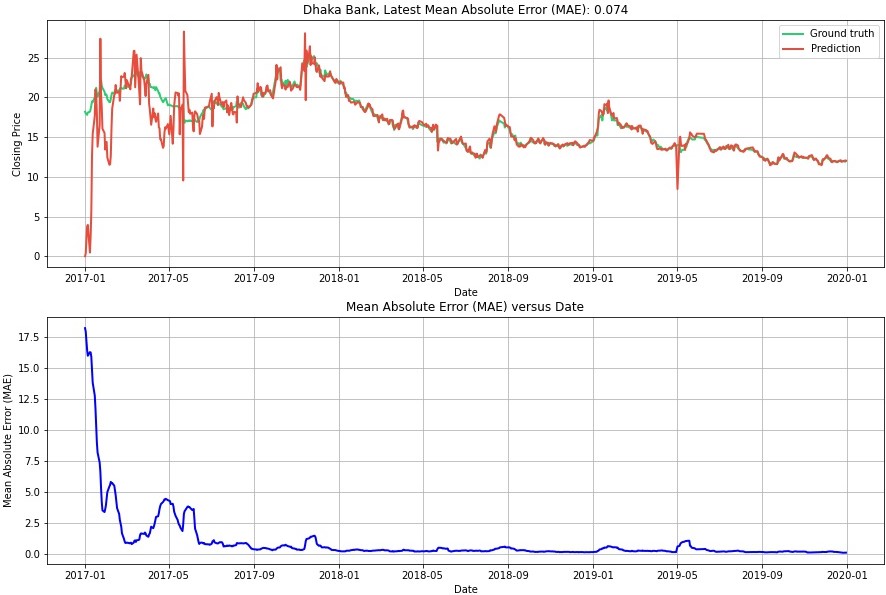
acme\_lab\_data\_path = "./data/dse-final-data/acme\_lab\_data.csv"

aramit\_cement\_data\_path = "./data/dse-final-data/aramit\_cement\_data.csv"

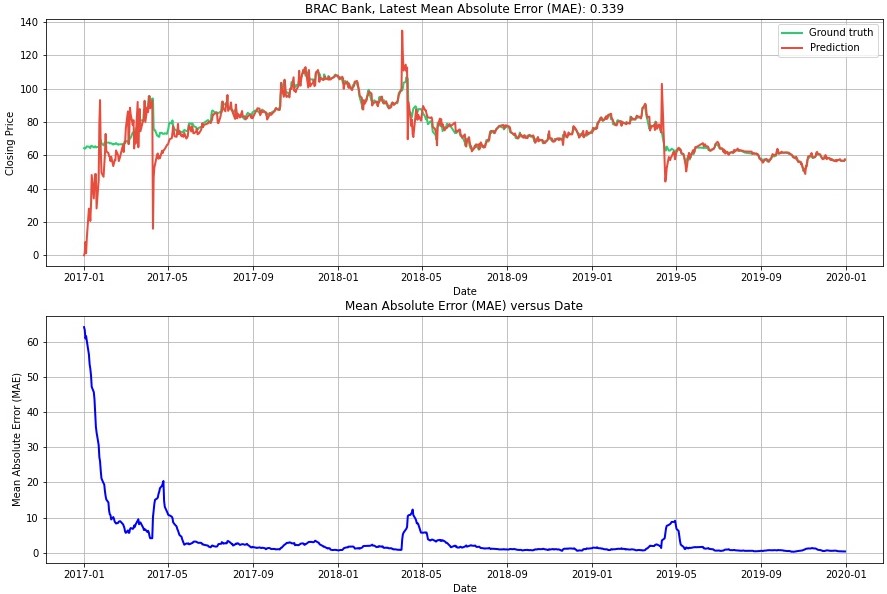
confidence\_cement\_data\_path = "./data/dse-final-data/confidence\_cement\_data.csv"

**Let’s evaluate our models.**

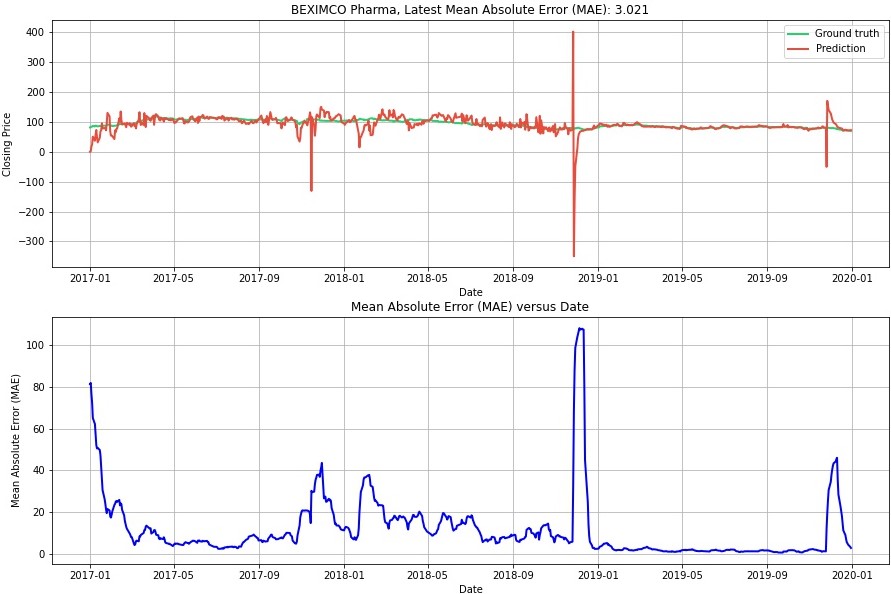
evaluate\_model(model\_1, "Dhaka Bank", dhaka\_bank\_data\_path)



evaluate\_model(model\_2, "BRAC Bank", brac\_bank\_data\_path)



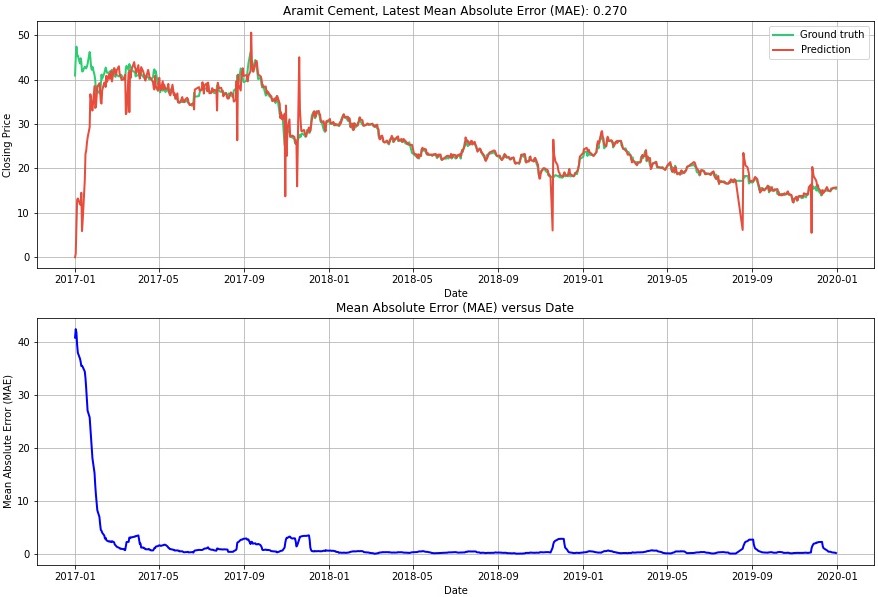
evaluate\_model(model\_3, "BEXIMCO Pharma", beximco\_pharma\_data\_path)



evaluate\_model(model\_4, "ACME Laboratories", acme\_lab\_data\_path)



evaluate\_model(model\_5, "Aramit Cement", aramit\_cement\_data\_path)



evaluate\_model(model\_6, "Confidence Cement", confidence\_cement\_data\_path)

