International Journal of Scientific Research in Science and Technology
Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)
doi : https://doi.org/10.32628/IJSRST

Machine Learning Model Approaches for Price Prediction in Coffee Market using Linear Regression, XGB, and LSTM Techniques

Tesyon Korjo Hwase[1], Abdul Joseph Fofanah[2]
*[1]Department of Software Engineering, Nankai University, MTT Consulting Architects and Engineers Plc., Addis Ababa, Ethiopia
*[2]Department of Mathematics and Computer Science, Milton Margai Technical University, Freetown, Western Area Rural, Sierra Leone

ABSTRACT

Article Info
Volume 8, Issue 6
Page Number : 10-48

Publication Issue
November-December-2021

Article History
Accepted : 01 Nov 2021
Published : 06 Nov 2021

Investors and other business persons have a desire to know about the future market price because, if the investors know about the future price of a certain commodity or stock it will help them to make appropriate business decisions and they can also get profit out of their investment. There are many previous researches that has been done on stock market predictions but there is no related research that has been done on Ethiopia commodity exchange (ECX). Performing future price prediction with better accuracy and performing comparative analysis between the algorithms for two of Ethiopia commodity exchange (ECX) items which are Coffee and Sesame as the research key objectives. Three different types of prediction algorithms to predict the future price, such as Linear Regression (LR), Extreme Gradient Boosting (XGB), Long Short-Term Memory (LSTM) was utilized. There are limited researches worked on price prediction of ECX items specifically, the idea of the price prediction on different Stock markets like New York stock market Exchange and other commodity market items prediction in order to develop our research in ECX was presented. The study apart from predicting the future price, comparative analysis was implemented between the prediction algorithms that we used based on their performance. Two different datasets from ECX: coffee and sesame were used. The reason for the utilization of these datasets is, the commodity items are the largest export items in Ethiopia which makes them very important for Ethiopian economy, and the different datasets helps us to get the advantage of evaluating the algorithms with different number of datasets, since sesame dataset has 7205 instances and coffee dataset has 1540 instances and both of them has 11 attributes. We build an android application in order two implement our algorithms on mobile applications and see if it is possible to implement the prediction algorithms on mobile platforms and make it easy and accessible to users. We call this mobile application Ethiopia Coffee Prices Predictor (ECPP). This application will be used to display the prediction result of Ethiopia Coffee price for short period and it is designed in the way to be user friendly. The programming environment used to implement the prediction algorithms is python, java programming language to design our android application and we used PHP to implement the API, and finally we used MySQL database in order to store information's online and make them accessible everywhere.

Keywords : Linear Regression (LR), Extreme Gradient Boosting (XGB), Long Short-Term Memory (LSTM), MySQL database, ECPP, ECX

I. INTRODUCTION

This paper aimed at price prediction of two of the largest coffee producers in Africa namely Ethiopian Commodity Market (ECX) items which are *Coffee* and *Sesame*, using different machine learning algorithms and doing a model comparison for different prediction techniques to find the most suitable one for our case. This section will mainly analyze the development background, the main research contents, and the structure of the paper for the price prediction of coffee, paving the way for algorithm design and implementation.

The prediction on price movements of financial time series can help investors avoid risks and obtain higher returns [1]. It is a hot yet challenging topic in the financial field. There are many types of research from various areas aiming to take on the challenge and it is an effective way to do the research using neural networks.

Coffee was first found in East Africa in the country that is called Ethiopia, Ethiopia is the origin of coffee and the largest coffee exporter and producer in Africa. Coffee is the major export item in Ethiopia and most of our income depends on it [2]. In the Ethiopian market sellers and buyers predict the price of coffee and other commodity items in a traditional way, in this method if they make a huge mistake, they will probably lose a lot of money, so because of this problem they use the commodity market to minimize the risk of their prediction. In the commodity market they use future contracts to minimize the loss for both sellers and buyers, A futures contract is an agreement between the buyer and seller to buy or sell a commodity at a particular price on a specified future date.

The market price is basically nonlinear in nature and the research on commodity or stock market is one of the most important issues in recent years. People invest their money in a certain commodity or stock based on some prediction this prediction can be made traditionally by people who have knowledge of the market. They give an analysis on what's going to happen next but these people need to be aware of the past history of the market and they must be well experienced in order to do good prediction on the market but still employing traditional methods like fundamental and technical analysis may not ensure the reliability of the prediction.

To predict the commodity market prices people search for methods and tools which will increase their profits while minimizing their risks. Buying and selling commodities in the commodity market is an act of gambling this misconception can be changed if we can be able to come up with a method that helps people to predict the price based on well-organized information or proof. Researchers use different methods to predict market prices some of them are data mining, regression models, Artificial Neural Networks (ANN), and so on.

In this paper, we will explore some of the machine learning algorithms to predict the price of coffee and sesame, and we will compare the efficiency of the algorithms for better results. We will design, implement, and analyze different algorithms in which we found the most effective ones for our case study. Apart from machine learning and RNN algorithms researchers try to apply different methods like data mining but for this paper, we will only be focused on the Linear Regression Model, Extreme Gradient Boosting Method, and Long Short-Term-Memory (LSTM).

Coffee in Ethiopia

Coffee production in Ethiopia is a longstanding tradition that dates back to many centuries. Ethiopia is where coffee arabica and the coffee plant originate. Coffee is now grown in various parts of the world and Ethiopia itself accounts for around 3-5% of the global coffee market. Coffee is important to the economy of Ethiopia; around 60% of foreign income for Ethiopia comes from coffee, according to researches with an estimated 15 million of the population relying on some aspect of coffee production or selling for their livelihood. In the year 2006, coffee exports brought in $350 million equivalent to 34% of Ethiopian that year's total exports [3].

Ethiopia is the world's fifth-largest producer of coffee and Africa's top producer with 384,000 metric tons in 2018[1]. Half of the coffee produced is consumed by Ethiopians and the country leads the continent in domestic consumption. Some of the major markets for Ethiopian coffee are the EU (about half of exports), East Asia (about a quarter), and North America. The income from coffee exports accounts for 10% of the annual government revenue, because of the large share the industry is given very high priority in Ethiopia.[3]

There are more than 1,000 known genetic varieties of coffee in Ethiopia [4]. This number is staggering especially considering the closest country in terms of variety is Colombia with about 30 known varieties. Even more shocking is that according to researches there are still thousands of varieties of Ethiopian coffee yet to be identified. A huge majority of Ethiopia's coffee is wild-harvested or grown on small farms, with only 20% commercially farmed.

II. RATIONALE FOR THE RESEARCH

The problem that we want to solve in this paper is to be able to predict the future price with better accuracy and better algorithm on two Ethiopian commodity exchange (ECX) items which are *Coffee* and *Sesame*. As far as we know there is no similar research done on ECX before. They traditionally perform the trading and sometimes with the help of professional economists.

When we talk about time series prediction in this specific case, we are talking about a timely pattern data set or commodity item. Because the commodity exchange is done in a timely sequenced manner, we call it time series data set. So, when we perform prediction using this dataset, we are going to take advantage of the time series data in order to be able to perform time series prediction.

Different researchers use different methods (Algorithm) to perform time series prediction, for example, RNN, different kinds of regression methods, Random Forest and other tree methods, RNN, and they also use classification algorithms. But, for this research paper, we will use LSTM, LR, and XGB specifically to perform time series prediction. In addition to doing comparative analysis, we also want to improve the accuracy of the models. When we talk about accuracy different researchers achieve different accuracy values from their prediction algorisms, the research was done by Chenhao Wang and Qiang Gao on High and Low Prices Prediction of Soybean Futures with LSTM Neural Network achieve 73.7% accuracy on closing data set using the LSTM algorithm and may other researchers achieve promising accuracy results on stock price prediction 70% and above MAPE. So, we want to perform the prediction with better accuracy by choosing the best feature and algorithm to work with.

When we see other country commodity markets, they do trade in a lot of commodity items apart from agricultural items like gold, oil, minerals, and so on. Researchers have done research on items like copper and oil. But until we did this research paper, we didn't find any research on predicting the future price of coffee specifically but we can use prediction on other commodity items as an example because the attribute of the data set is almost the same. Following the above analysis, we put forward the following research questions: which one is the best algorithm to predict ECX market? how to achieve better accuracy? and how to implement prediction algorithms on a mobile device and make it accessible for everyone?

Consequently, the researcher proposed the following challenges:

- Accuracy is the main problem when it comes to price prediction
- Traditional price prediction system in ECX there is no modern and accessible price prediction system
- Implementing prediction algorithms in mobile devices is quite a challenging task

III. PRICE PREDICTION USING MACHINE LEARNING MODELS

Prediction tools are powered by several different models and algorithms that can be applied to a wide range of use cases. Being able to decide which predictive modelling techniques are best for your company is key to getting the most out of a predictive analytics solution and leveraging data to make insightful decisions [5].

In this 21st century, we hear the term "machine learning" a lot, usually in the context of predictive analysis and artificial intelligence. Machine learning is, more or less, a way for computers to learn things without being specifically programmed for everything. There are different ways to predict a price using Artificial intelligence but for this paper, we will focus only on the following: *regression method (linear regression), tree-based method (extreme gradient boosting), and neural networks (long short-term memory (LSTM)).*

The key component of every result of the algorithm is the accuracy it delivers. It should be according to our needs and as stated earlier. There are some standard methods to calculate accuracy in machine learning, some are as follows: *R2 value of the model, adjusted R2 value, MAPE, and RMSE Value Confusion matrix for classification problems, and many more.*

A. Linear Regression

We choose this algorithm because of its simplicity and because it serves our purpose in taking an algorithm from the regression method for prediction. Linear regression is a very commonly used technique for data analysis and forecasting. It essentially uses the key features to predict relations between variables based on their dependencies on other features [6].

In linear regression, we build a model (equation) based on our data. then we can use this model to make predictions about one variable based on particular values of the other variable. The variable that we are going to making predictions about is called the dependent variable (also commonly referred to as y, the response variable, or the criterion variable). The variable that we are using to make these predictions is called the independent variable (also commonly referred to as x, the explanatory variable, or the predictor variable) [7].

This is, in fact, the line that we were eyeballing in the opening section of the module. Using linear regression, we will be able to calculate the best fitting line, called the regression line [7]. Consequently, the criterion for the best fitting line is the line that minimizes the sum of our square errors which is the sum of the differences between the plotted points and our line as illustrated below.

Its disadvantage is that there is a tendency for the model to "overfit" that is, for the model to adapt to the data on which it has been trained at the ability to generalize to previously unseen data. For this reason, linear in machine learning is often "regularized," which means the model has certain penalties to prevent overfitting [8].

B. Extreme Gradient Boosting (XGBoost)

A decision tree is graph-based that uses a branching method to show each possible outcome of a decision. It is possible to represent all possible outcomes in a decision tree. In ML, the branches used are binary yes or no answers.

In order to train a decision tree, we take the train dataset (that is, the data set that we use to train the model) and find which attribute best to "split" the train set with regards to the target. After we did the first split, we have two subsets that are the best at predicting if we only know that first attribute. Then we can do iteration on the second-best attribute for each subset and re-split each subset, continuing until we have used enough of the attributes to satisfy our needs.

Extreme Gradient boosting (XGBoost) is built on the principles of gradient boosting. XGB is made in the way to accomplish the extreme of the computation limits of machines to provide better accuracy, speed, and efficiency. This algorithm has different names some of them are gradient boosting, multiple additive regression trees, stochastic gradient boosting, or gradient boosting machines.

The way gradient boosting work is quite different, in which the model that is created will predict the error of the prior models, then it will add them together to make the final prediction. The reason why It got the name gradient boosting is that it uses a gradient descent algorithm to minimize the loss when adding new models. XGB is one of the implementations of the Gradient Boosting concept, but what makes XGBoost unique is that it [9].

This method can be used for both regression and classification problems, Gradient boosting gives a prediction model by combining a weak prediction model or decision trees [10]. It constructs the model in a sequence fashion as other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

Random forest is made from "weak" decision trees. The huge difference is that in gradient boosting, the trees are trained one after another. Each sequential tree is trained primarily with data that had been incorrectly predicted by previous trees. This helps the gradient boost to gradually focus less on the easy-to-predict cases and more on difficult cases. So, models are added sequentially until no further improvements can be made. one of the popular examples is the AdaBoost algorithm that weights data points that are hard to predict.

Boosting usually takes slower steps, making predictors sequentially instead of independently. It reportedly leverages the patterns in residuals, strengthens the model with weak predictions, and makes it better. XGBoost gave a prediction error almost ten times lower than boosting or random forest in my case [11].

C. Long-Short Term Memory (LSTM)

Neural networks indicate a biological phenomenon comprised of interconnected neurons that exchange messages with each other. The idea has now adapted to the world of machine learning and is called ANN (Artificial Neural Networks). Deep learning, can be done with several layers of neural networks put one after the other.

ANNs are a family of models that perform better and inefficient way when it comes to big datasets [12]. Other algorithms cannot handle extremely complex tasks, such as image recognition, as well as neural networks. But, just like the human brain, it takes a very long time to train the model, and it requires a lot of power.

The long short-term memory (LSTM) neural network is a special recurrent neural network. It was designed by Hochreiter and Schrnidhuber in 1977 aiming to solve the problems of vanishing and exploding gradients [13]. The key to the LSTM solutions of the technical problems is the specific internal structure of the units used in the model. The units called "gates " allow for weights adjustments as well as truncations of the gradient when its information is not necessary. In recent years LSTM is becoming popular to solve a large variety of problems very well.

LSTMs neural network is made to avoid the long-term dependency problem. Remembering and keeping information for long periods of time is practically their default behaviour, not something they struggle to learn. The reason why we chose LSTM specifically than Recurrent Neural Network (RRN) is that one of the appeals of RNNs is the idea that they may be able to connect previous information to the present task, like using previous video frames might inform the understanding of the present frame. unlike the practice, in theory, RNNs are absolutely capable of handling such “long-term dependencies.” A human could carefully pick parameters for them to solve the toy problems of this form. But, in practice, RNNs can't seem to be able to learn them. The problem was discussed in depth by Hochreiter (1991) [German] and Bengio, et al. (1994), who found some pretty fundamental reasons why it might be difficult but LSTMs don’t have this problem [14].

LSTMs neural network help preserve the error that can be backpropagated through time and layers. LSTM lets the recurrent network learn over many time steps by keeping a more constant error. This is the challenge to ML and AI since algorithms are frequently confronted by environments where reward signals are sparse and delayed, such as life itself.

Research Objectives

The main objectives of this research are:

1. Performing future price prediction on two of ECX items which are coffee and Sesame using Linear Regression, XGB, and LSTM. Given prices and other features for the last N days, we do prediction for day N+1.
2. Performing comparative analysis based on the output we get from the algorithms that we use.
3. Building an android mobile application to implement our algorithms and make it user friendly.

IV. PROPOSED RESEARCH METHODOLOGY

Research Design

We use different methodology to achieve our vision for this paper some of them are the following:

- The first step for every algorithm to function properly is data preprocessing, the data that we get from Ethiopian commodity market exchange is row data which contain all commodity items because of this problem we did a data processing to get the data that is only necessary to our algorithms.
- The second step after we finish data preprocessing, we did data visualization using different python libraries which is very important in analyzing the market behavior, market influences and pattern.
- Predicting the next day price for coffee and sesame dataset using linear regression, Extreme Gradient Boosting, and Long Short-Term Memory (LSTM).
- Evaluate their performance and give analysis for next day prediction according to the loss function of the model and their performance.
- Finally, we will give a comparative analysis of all the results that we get from these three algorithms and choosing the better-performed algorithm.

In addition to the predictive algorithms, we are also going to build an android mobile application in order to implement our models on the mobile platform and to make the predictive algorithms accessible to all who have a desire to use it.

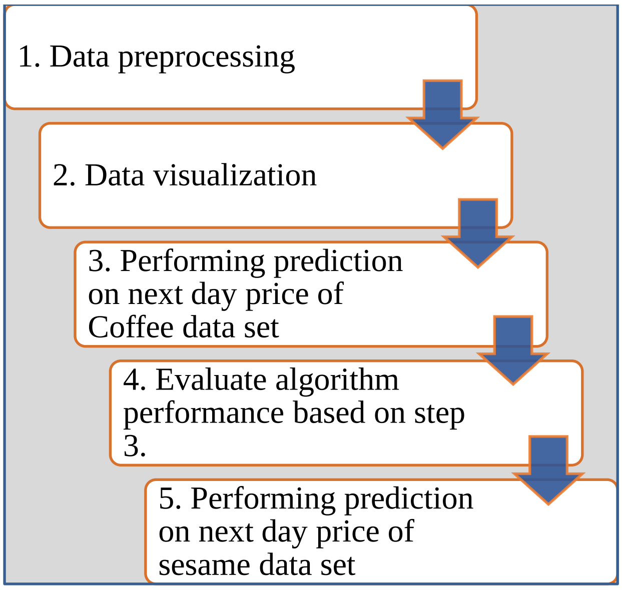

1. Data preprocessing
2. Data visualization
3. Performing prediction on next day price of Coffee data set
4. Evaluate algorithm performance based on step 3.
5. Performing prediction on next day price of sesame data set

Figure 1: General and specific methodology diagram

Research Method

We will use a machine-learning algorithm to predict the future Coffee and sesame market price for exchange by using open-source libraries and modifying pre-existing algorithms to help make this unpredictable format of business a little more predictable.

The price prediction for the Ethiopian coffee market involves many components and algorithms. One from the regression model, one from the tree-based model, and one from the recurrent neural network model. We use these selected algorithms to find the best prediction model for our dataset.

We used data set from the Ethiopian Commodity Exchange Market (ECX) from 2012 to 2019 which has 11 attributes.

V. RELATED WORKS AND THEORIES

Predicting the price of an item is an act of trying to determine the future value of that specific item. If we are able to predict stocks or items future prices could yield a significant profit. There are many types of research that have been done on stock market prediction which we want to use almost similar principles for our study case which is Ethiopian coffee market prediction. Because there is not much research done on the specific topic we are doing, we learn how to predict price by reading researches that have been done on stock market prediction and other price predictions because the concept and the behavior of the data that we are using are almost similar to the stock market and commodity market data. This chapter shows how the topic of this work fits into the previous research and the history behind it and also the relevant theory and technology that is similar to our system.

a. Supervised Learning-Based Predictions

Several algorithms have been used in price prediction such as SVM, Neural Network, Linear Discriminant

Analysis, Logistic Regression, Linear Regression, KNN, and Naive Bayesian Classifier. according to findings, logistic regression was one of the best with a success rate of 55.65%. Dai and Zhang (2013) used the training data set from 3M Stock data [15]. The dataset contains daily stock information from 1/9/2008 to 11/8/2013 (1471 data points). Many algorithms were chosen to train the prediction system. These algorithms include Logistic Regression, Quadratic Discriminant Analysis, and SVM. The algorithms were applied to the next day model which predicted the outcome of the stock price on the next day and the long-term model, which predicted the outcome of the stock price for the next x days. The next day prediction model produced accuracy results ranging from 45% to 58%. SVM achieved the highest accuracy of 79.3%, For the long-term prediction where the time window has taken, was 44. In one of the published papers [8] on ANN (artificial neural network), authors used for the forecast the direction of the Japanese stock market gave an accuracy of 81.3%. In the published paper on [16] Random Forest (RF), 81.27%. In the published paper on Random Forest, an ensemble technique is used to predict the stock market prices (Trend Up or Down) and returned the highest accuracy of 79% for the direction of movement in the daily TSE (Tehran Stock Exchange) index [17].

Sneh Kalra and Jay Shankar Prasad in their 2019 conference paper, this paper reveals the relationship between movements in stock markets and financial news. they study the relationship of social media, news articles, and financial reports on future stock returns and the need to find appropriate forecasting methods to find the best results [16]. according to this paper, Positive News encourages individuals to buy and sell stocks respectively. New products and acquisitions, better quality earnings reports, overall economic and political indicators increase the demands and stock price. In contrast, negative news leads to a decrease in demands and stock prices and causes individuals to sell stocks. For our case, we take the volume as a factor that affects the price of the coffee.

Nicole Powell, Simon Y. Foo, and Mark Weatherspoon in 2008 publish a conference paper which states about Supervised and Unsupervised Methods for Stock Trend Forecasting. They use supervised learning, unsupervised including the decision on whether the stock's price classification, k-means clustering, support vector machine algorithms to do prediction and comparative analysis [6]. The aim of the paper is to do a classification model to classify stock prices as either high or low. In our case, we only want to focus on supervised learning other than unsupervised learning.

In June 2018, Siyuan Liu, Guangzhong Liao, and Yifan Ding in their conference paper called “Stock Transaction Prediction Modeling and Analysis Based on LSTM” stated that LSTM (Term Memory Long-Short) is a kind of time recurrent neural network, which is suitable for processing and predicting the important events of interval and a long delay in time series [18]. Based on this temporal characteristic of stock and LSTM neural network algorithm, this paper uses the LSTM recurrent neural networks to filter, extract feature value and analyze the stock data, and set up the prediction model of the corresponding stock transaction. They use a different method to implement their work, the data processing is the main part. They use two different methods to do data pre-processing the first one is calculating the stock’s MA, EMA index by the closing price data. Secondly, they preprocess the correlation index of stock data according to the following methods [18].

$$Oc = (open - close)/open$$
$$Oh = (open - high)/high$$

In Nov. 2018, Chenhao Wang and Qiang Gao in their conference paper that is called “High and Low Prices

Prediction of Soybean Futures with LSTM Neural Network" predict soybean prices using high and low prices, from the Dalian Commodity Exchange dataset [19]. They use high and low prices because this one has low noise when it compares to closing price. they take low prices as the result of maximum and minimum filtering of trading prices during a certain period. According to their finding, the noise of high and low prices is lower compared with closing prices and it is easier to predict movements of the high and low prices. To compare the efficiency, they use mean absolute error (MAE) and trend accuracy to evaluate the performance of this model.

Their final results show that the accuracy of predictions on the trends of high and low prices is higher than closing prices. The LSTM model gets about 70% of trend accuracy and simple strategies were applied to test the feasibility. They get ideal profits through the model. And the model performs better when the high and low prices have high volatilities.

Our aim for this paper is almost similar to Chenhao Wang and Qiang Gao but other than LSTM we use regression model and tree-based models to explore more options.

In Sept. 2019 Andrei Ioaneş and Radu Tîrnovan in their conference paper that is called "Energy Price Prediction on the Romanian Market using Long Short-Term Memory Networks" predict the energy price for the Romania market using the LSTM method [20]. Long Shot-Term Memory networks appeared here as a solution to sequential data problems in which individual elements are related either in time or space. The prediction is for long- and short-time intervals starting from the next day to a month, based on this the result that they get is interesting, for short intervals of time (24h) the model achieves values with a maximum 5% accuracy while increasing the time span results in an accuracy drop by more than 20% and reliability becomes the main obstacle. The model is unable to predict "irrational behavior" when the price displays an evolution completely different from the trend set in previous epochs due to unpredictable events. subsequently, they come to the conclusion that Thus, the results obtained are found to be satisfactory when and the use of LSTM and Keras library demonstrates the effectiveness and applicability of the proposed methodology. Using a larger data set input and training set with weights being defined more accurately might considerably enhance the model. For our paper when we use LSTM our aim is to improve the efficiency of this algorithm for a long and short period of time.

S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, propose a deep learning-based formalization for stock price prediction. It seems that deep neural network architectures are capable of capturing hidden dynamics and are able to make predictions. They trained the model using the data of Infosys and were able to predict the stock price of Infosys, TCS, and Cipla. This indicates the system that they proposed is capable of identifying some interrelation within the data. Also, it is evident from the results that, CNN architecture is capable of identifying the changes in trends [21].

b. Gradient Boosting and AdaBoost Algorithm

Before we start discussing extreme gradient boosting it is necessary to understand gradient boosting and AdaBoost Algorithms. Boosting is a method used to convert weak learners into strong learners. Here, each new tree is a fit on a modified version of the original dataset. GBA can be most easily explained by first introducing the AdaBoost Algorithm [22]. The AdaBoost starts by training a decision tree in which each observation is assigned an equal weight.

Gradient Boosting usually trains models in a gradual, additive, and sequential manner. The main difference between AdaBoost and Gradient Boosting Algorithm is how the two algorithms identify the shortcomings of weak learners (e.g., decision trees). The AdaBoost algorithm identifies the Problems by using high-weight data points, in the other side gradient boosting performs the same by using gradients in the loss function.

The loss function is a measure of how good our model's coefficients are at fitting the underlying data. the basic understanding of loss function would depend on what we are trying to optimize. to take an example let say we are trying to predict the sales prices by using a regression, then the loss function would be based on the error between true and predicted house prices. Again, if we aim to classify credit defaults, then the loss function would be a measure of how good our predictive model is at classifying bad loans. The initiation for using gradient boosting is that it allows one to optimize a user-specified cost function instead of a loss function that usually offers less control and does not essentially correspond with real-world applications [23].

Y. Zhang and A. Haghani explain gradient boosting as Tree-based ensemble methods that reached celebrity status in the prediction field. By mixing simple regression trees with 'poor' performance they usually produce high prediction accuracy [24]. Unlike other machine learning methods, tree-based methods provide a result that is interpretable, while requiring little data preprocessing, are able to handle different types of predictor variables, and can fit complex nonlinear relationships [24] [23].

c. Extreme Gradient Boosting

There are many machine learning techniques in the wild, but extreme gradient boosting (XGBoost) is one of the most popular. Gradient boosting is a process to convert weak learners to strong learners, in an iterative fashion. The name XGBoost refers to the engineering goal to push the limit of computational resources for boosted tree algorithms. Since its introduction in 2014, XGB has shown to be a very powerful machine learning technique and is usually the go-to algorithm in many Machine Learning competitions [11].

The decision tree constructs a tree that is used for classification and regression, but trees that are grown deep to learn highly irregular patterns tend to over-fit the training sets [25]. A small change in the data may cause the tree to grow in a completely different manner. the reason is that decision trees have very low bias and high variance.

Extreme Gradient Boosting has been used for prediction for quite a long time now including stock price and so many other price predictions [26]. Bellow we will introduce different fields that used the Extreme Gradient Boosting method for prediction.

On 24 October 2019 Yuanxiang WANG, Qing DENG, Fushun WEN, Hongyu ZHOU, Fuyan Liu, and Xiaoyong Yang stated in their work about predicting price for High Voltage Transmission Projects using Extreme Gradient Boosting [27]. The purpose of this study is because it is highly demanded to establish an accurate cost prediction method for an Ultra High Voltage (UHV) transmission project to ensure investment efficiency. The traditional method or of determining the project cost according to the project quota and estimated budget can order to build t make full use of historical data of similar projects and cannot well meet the project investment needs. Especially in the middle and late stages of a project, it is very difficult for the cost management department to judge the advancement and rationality of the project design scheme and construction technology by the project quota and estimated budget. In addition to Extreme Gradient Boosting, they also use a genetic algorithm-based support vector machine (GA-SVM) in order to build a more accurate combinatorial forecasting method.

In July 2019 Jerome Cary Beltran, Paolo Valdez, and Prospero Naval Jr., Ph.D. in their conference paper that is called" Predicting Protein-Protein Interactions based on Biological Information using Extreme Gradient Boosting" in their prediction protein-protein interactions (PPIs), eXtreme gradient boosting (XGBoost) outperform Support Vector Machine (SVM) and Random Forest (RF) in terms of accuracy, specificity, Matthews correlation coefficient (MCC), Area Under the Curve (AUC) and F1 Score. According to their finding, XGBoost performs better than RF in all metrics except sensitivity. During model training, XGBoost builds trees one at a time, where each new tree helps to correct errors made by the previously trained tree. Thereby, a more expressive model is built at the cost of more computational time [19].

On Oct 27, 2019, Yibin Ng in his article that is called 'Forecasting Stock Prices using XGBoost' he writes about extreme gradient boosting for the stock market in a very good way. The aim of the article is to predict the daily adjusted closing prices of Vanguard Total Stock Market ETF (VTI), using data from the previous N days. In this experiment, he uses 6 years of historical prices for VTI from 2013–01–02 to 2018–12–28, which can be easily downloaded from yahoo finance. He predicts for 21 days in the future because there are about 21 trading days in a month, excluding weekends for that he uses a technique that is known as recursive forecasting and he used the moving window validation method to perform hyperparameter tuning his findings, he stated that using XGBoost with or without the date features gives better performance over the Last Value method. Interestingly, omitting the date features gives a slightly lower RMSE than including the date feature he also found, the date features have a low correlation with the target variable and likely do not help the model much [28].

Shubharthi Dey, Yash Kumar, Snehanshu Saha, and Suryoday Basak Predict the direction of the stock market price using Xtreme Gradient Boosting and they published a paper that is called 'Forecasting to Classification: Predicting the direction of the stock market price using Xtreme Gradient Boosting' the main aim of this paper is to predict the rise and fall of the stock market. The measuring parameter they used is +1 for indicating the rise in stock valuation in the future and -1 to indicate the fall in the prices [17]. The model is found to be robust in predicting the future direction of the stock movement. The robustness of their model has been evaluated by calculating various parameters such as accuracy, precision, recall, and specificity. For all the datasets we have used i.e., Apple and Yahoo, we were able to achieve accuracy in the range 87- 99% for long-term prediction. ROC curves were also plotted to evaluate their model. The curves demonstrate the fidelity of their model graphically. Their model can be used for devising new strategies for trading or to perform stock portfolio management, changing stocks according to trends prediction. In the future, they also could build boosted tree models to predict trends for a really short time window in terms of hours or minutes. Different ML algorithms can also be checked for their robustness in stock prediction. They also recommend exploration of the application of Deep Learning practices in Stock Forecasting involving learning weight coefficients on large, directed, and layered graphs.

d. Regression Technique

Regression is a method of modelling a target value based on the independent variable or predictor [29]. This method is mostly used for prediction and finding out the cause-and-effect relationship between variables [30]. Regression techniques most of the time differ depending on the number of independent variables and the type of relationship between the independent and dependent variables [31].

In 2019 explore how to predict the stock price using supervised machine learning [32]. This paper is limited to only supervised machine learning and tries to explain only the fundamentals of this complex process. They specifically focused on the Linear regression method and improving its efficiency [33]. For our case study, we also use linear regression and also other supervised learning algorithms.

Yahya Eru Cakra and Bayu Distiawan Trisedya, done their research on Price Prediction using Linear Regression based on Sentiment Analysis. The purpose of this research was to predict the Indonesian stock market using simple sentiment analysis. Naïve Bayes and Random Forest models are used to classify tweets to calculate sentiment regarding a company [34]. The output that they get from sentiment analysis will be used to predict the company stock price. They used a linear regression method to build the prediction model and their experiment shows that prediction models using previous stock price and hybrid feature as predictor gives the best prediction with 0.9989 and 0.9983 coefficient of determination. G. A. F. Seber and A. J. Lee also explains how linear regression works and the math's behind it in their book that is called Linear Regression Analysis [35].

In December 2012 Han Lock Siew and Md Jan Nordin state in their paper Regression Techniques for the Prediction of Stock Price Trend, they examine the theory and practice of regression techniques for prediction of a stock price trend by using a transformed data set in ordinal data format [19].

In 2016 Kavitha S, Varuna S, and Ramya R In their paper. linear regression and support vector regression model is compared using the training data set in order to use the correct model for better prediction and accuracy [36]. According to them, there are many technologies used in data analytics but predictive analytics is the one that uses machine learning algorithms and statistical analysis for future prediction. At the end of their paper, they conclude that data Analytics and business intelligence play a major role in the current competitive market. In the case of analyzing a time series multivariate analysis, an efficient data model should be used for accurate results. For our research case we are not predicting the stock market we are predicting coffee price specifically but it is possible to use the same principle as predicting the stock market because their behavior is almost similar.

Commodity Markets and Future Price Predictions

The Ethiopia Commodity Exchange (ECX) is a commodities exchange established in 2008 in Ethiopia. The main objective of ECX was "to ensure the development of an efficient modern trading system" that would "protect the rights and benefits of sellers, buyers, intermediaries, and the general public [29]. The ECX is owned by shareholders that are not members [30]. ECX offers customers access to trading for five different agricultural products, each of which has many varieties: *Coffee is the number one traded item, sesame is the second-largest traded item, haricot beans, maize (corn), and wheat.*

Sesame markets in Ethiopia are highly linked with the international market and highly volatile following changes in the supply and demand in the international arena. As can be seen in the following Figure, prices of sesame peaked in the first quarter of 2008 and are still relatively high in 2009.

A. Beber and J. Piana, study the links between expectations, fundamentals, and asset returns using the rich empirical setup offered by commodity markets. They find that survey-based expectations predict future fundamentals but are not significant predictors of future returns. Expectations of returns are related to the slope of the commodity futures term structure and with trading flows. Furthermore, dispersion in analysts' forecasts helps in explaining the options implied volatility risk premium.

Interestingly, time-series momentum exhibits a strong negative relation with survey-based expectations [31]. They rationalize these findings using a simple model with heterogeneous beliefs, where professional forecasters can still have rational expectations.

S. Madria, P. Fournier-Viger, S. Chaudhary, and P. K. Reddy, study the design and implementation of Market Intelligence System Proof of Concept (PoC) using available datasets for a few agricultural commodities. This System Proof of (PoC) takes daily market price and weather data as input transforms it into information and generates actionable intelligence by applying forecasting and deep learning techniques. The system provides trend analysis for short as well as long-term commodity price prediction and market selection as insights for farmers [32]. The Auto-Regressive Integrated Moving Average (ARIMA) forecasting technique and Recurrent Neural Network (RNN) deep learning techniques are applied for short-term and long-term agricultural commodity price prediction respectively. The final study results demonstrate the intended utility of forecasting and deep learning techniques for generating a Market Intelligence System.

Finally, the paper concludes with the benefits of a comprehensive Market Intelligence system, challenges, and future work. D. Bakas and A. Triantafyllou, examine the predictive power of macroeconomic uncertainty on the volatility of agricultural, energy, and metals commodity markets. In their research, they found that the latent macroeconomic uncertainty measure of Jurado et al. which is a usual volatility forecasting factor for the commodity market which provides volatility predictions for forecasting horizons up to twelve months ahead. Their results indicate that the forecasting power of macroeconomic uncertainty is higher when predicting the volatility of energy commodities [33]. Their findings also show that higher macroeconomic uncertainty is associated with large volatility episodes subsequently observed in all commodity markets.

The predictive power of the unobservable macroeconomic uncertainty factor remains robust to the inclusion of observable economic uncertainty measures, historical commodity price volatility, stock-market realized, and other macroeconomic variables that are highly related to the production process and also the mechanics of commodity markets (see Figure 2).

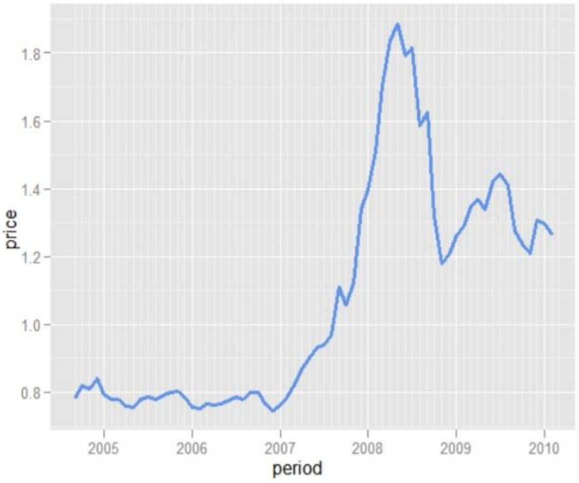


Figure 2: Sesame in the commodity market

VI. PROCEDURES AND REQUIREMENT ANALYSIS

This section contains information on the key technologies, algorithms, and the analysis of the required strategy decisions that went into this paper. The architecture patterns are outlined and explained. The entire algorithm and system are shown in a few diagrams and the subsystems are explained. Finally, the design patterns used in the project are discussed.

Research Purpose

Coffee is the big market in Ethiopian which covers 65% of Ethiopian foreign currency income and it is the biggest export item from Ethiopia to the rest of the world and sesame is also the largest export item next to coffee, so it is very important to do technological advancement on the traditional trading system.

When investors and farmers make a business plan it is very important to acknowledge price change and move accordingly, but the problem is in order to guess or predict the future price the person needs to be knowledgeable and should be aware of the market behaviour and previous events in the market. So, our purpose here is to be able to find a machine learning algorithm solution that can analyze historical data and be able to do future price predictions.

The largest coffee market in Ethiopia is the Ethiopian Commodity Market (ECX), the majority of coffee trading is through this market. In this commodity market, the buyers and sellers have to sign a contract that is called a futures contract, shortly it means settling the price for the future trading date which they do not know what the price will be but they need to come to an agreement by predicting the future price. So, our purpose is to provide a technological solution to help them to understand or give them an idea of how the market will behave for future dates. Apart from doing prediction, our purpose is to do a comparative analysis between three supervised learning prediction models to understand which one will perform better for our research case.

Finally, we will build a mobile application user interface to make it easy to use for both buyers and sellers if they are interested in doing predictions using our prediction model. In addition, we strongly believe that doing this user interface will initiate other researchers to have an interest in improving the system and do more advancement.

Scope of the Research

Knowing Ethiopian coffee market price is very important to the investors and farmers who produce coffee because it will directly affect them. If they get the idea of whether the price will go up or down it will make it easy for them to make a business plan, so in order to achieve that in this paper we will be predicting the Ethiopian coffee market price as well as sesame market price for a short period of time.

We will use three algorithms to predict the price the first one is linear regression, the second one is Extreme gradient boosting and the final one is Long Short-Term Memory (LSTM). In this paper Apart from predicting the future price we will compare the performance of these three algorithms and their behavior for our case study. So, the main objective of this paper will be predicting future prices, performing comparative analysis, and building a mobile application for algorithms so it can be applicable for anybody who has an interest in seeing what the machine is predicting.

The goal is to introduce a new way of predicting price for my country because we are a developing country and we are not much familiar with machine learning and AI use and its wide range of applications, so this study will be one way of showing its wide range application and it will help people to have the interest to do more discovery and study.

Overall Description

The first step is to predict the market price using the algorithms that we mention above the detailed design will be explained in the next section, after predicting and getting the prediction value the next step will be to do a comparative analysis. We will do the comparative analysis based on the performance of the algorithm for a short-term prediction.

After we finish the comparative analysis and output our model result in CSV format the next step will be putting the result, we got in our online server so that we can update and fetch data easily so controlling and updating the prediction data will be the job of the administrator of the database.

The final step is to make a connection between the online database and the mobile application using API so the application will be able to fetch updated data from the database. The mobile application is designed to be user-friendly and be able to display the prediction value for both short- and long-term prediction in a graph format and again it also gives a business plan suggestion depending on the prediction result that we get from the best-performed algorithm.
In addition, the mobile application has the following Activities: *welcome page, home page to choose coffee type so they can be directed specifically to selected type prediction, an activity which displays the prediction graph for the previous and future day, and prediction display activity for each coffee type* (see Figure 3-left).

Algorithms Description and Architecture

All algorithms have their own architecture and working principle. For our case, the algorithms that we implement are linear regression, Extreme gradient boosting, and LSTM. Each of them has its own architecture and working principles but their purpose here is the same we are going to use all of them for one aim which is predicting future price because their architecture and working principle are different from the result, we get from each of them is different including their efficiency toward prediction.

The core idea behind LSTMs

LSTM is a very special kind of recurrent neural network (RNN) that works, for many tasks, much better than the standard version, capable of learning long-term dependencies. Most of the exciting results are based on recurrent neural networks that are achieved with them.

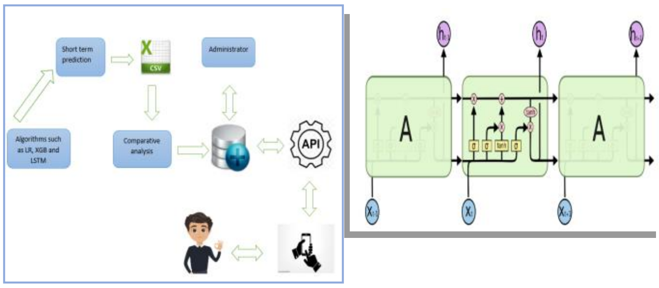


Figure 3: Overall description diagram (left) and the repeating module in an LSTM contains four interacting layers (right).

In the previous recurrent neural network, which is during the gradient back-propagation phase, there is a high possibility that gradient signal can end up being multiplied a large number of times which is the same as the number of timesteps by the weight matrix associated with the connections between the neurons of the recurrent hidden layer [34]. This shows that the magnitude of weights in the transition matrix can have a massive impact on the learning process. Here we need a notice that weights in this matrix are small it can lead to a situation called vanishing gradients when the gradient signal gets very small that learning either becomes too slow or stops working altogether. It can also make the task more difficult of learning long-term dependencies in the data. Conversely, if the weights in this matrix are large or, again, more formally, if the leading eigenvalue of the weight matrix is larger than 1.0, it can lead to a situation where the gradient signal is too large that it can cause learning to diverge. These are also referred to as exploding gradients. This is the main motivation behind the LSTM model which introduces a new structure called a memory cell.

A memory cell is composed of four main elements which are: an input gate, a neuron with a self-recurrent connection which is a connection to itself, a forget gate, and an output gate. The self-recurrent connection is going to weigh 1 and ensures that barring any outside interference the state of a memory cell can remain constant from a one-time step to another. The gates are used to modulate the interactions between the memory cell itself and its environment [35]. The input gate can let the incoming signals alter the state of the memory cell or block it and also the output gate can let the state of the memory cell in order to have an effect on other neurons or prevent it. The forget gate can modulate the memory cell's self-recurrent connection, allowing the cell to remember or forget its previous state, as it's necessary (see Figure 3-right).

LSTM Architecture for a next day prediction

Figure 4 shows the LSTM architecture which we will be using. We will use two layers of LSTM modules, and a dropout layer in-between to avoid over-fitting.

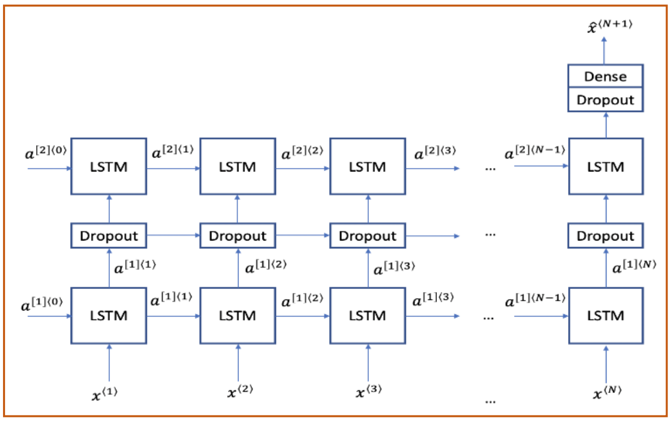

$\hat{x}^{\langle N+1\rangle}$
Dense
Dropout
$a^{[2]\langle 0\rangle}$
LSTM
$a^{[2]\langle 1\rangle}$
LSTM
$a^{[2]\langle 2\rangle}$
LSTM
$a^{[2]\langle 3\rangle}$
...
$a^{[2]\langle N-1\rangle}$
LSTM
Dropout
Dropout
Dropout
...
Dropout
$a^{[1]\langle 1\rangle}$
$a^{[1]\langle 2\rangle}$
$a^{[1]\langle 3\rangle}$
$a^{[1]\langle N\rangle}$
$a^{[1]\langle 0\rangle}$
LSTM
$a^{[1]\langle 1\rangle}$
LSTM
$a^{[1]\langle 2\rangle}$
LSTM
$a^{[1]\langle 3\rangle}$
...
$a^{[1]\langle N-1\rangle}$
LSTM
$x^{\langle 1\rangle}$
$x^{\langle 2\rangle}$
$x^{\langle 3\rangle}$
...
$x^{\langle N\rangle}$

Figure 4: LSTM architecture

After tuning the parameter in different way, the following architecture (Figure 5) with the specified number of LSTM layer, dropout, dense gave us a better result in terms of performance.

Prediction using extreme gradient boosting (XGB)

Core idea behind XGBoost

XGBoost is a decision tree-based Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data like images, text, etc. artificial neural networks used to outperform all other algorithms or frameworks. Decision tree-based algorithms are considered best-in-class right now for small-to-medium structured or tabular data. The following chart shows the evolution of tree-based algorithms over the years [37].

These two tree-based methods which are, Extreme Gradient Boosting (XGB) and Gradient Boosting Machines (GBMs) use the principle of boosting weak learners using gradient descent architecture. XGB got improved on the base of the GBM framework through systems optimization and algorithmic enhancements.

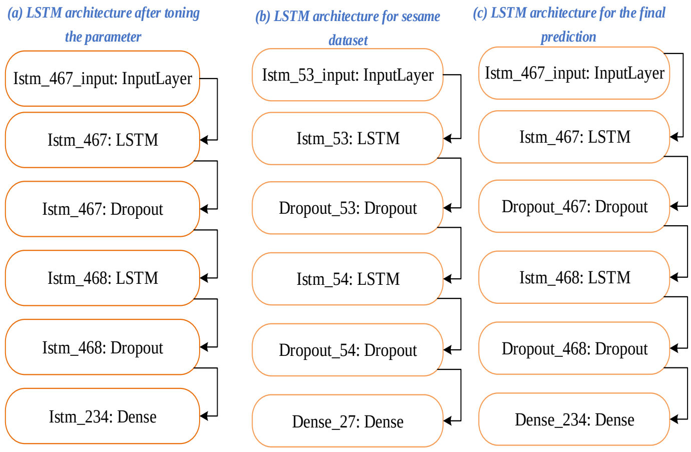


Figure 5: LSTM architecture after toning the parameter (a), LSTM architecture for sesame dataset (b), and LSTM architecture for the final prediction (c)

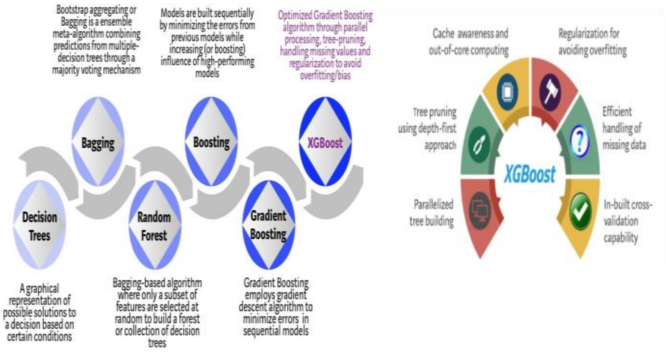

Bootstrap aggregating or Bagging is a ensemble meta-algorithm combining predictions from multiple-decision trees through a majority voting mechanism
Models are built sequentially by minimizing the errors from previous models while increasing (or boosting) influence of high-performing models
Optimized Gradient Boosting algorithm through parallel processing, tree-pruning, handling missing values and regularization to avoid overfitting/bias
Bagging
Boosting
XGBoost
Decision Trees
Random Forest
Gradient Boosting
A graphical representation of possible solutions to a decision based on certain conditions
Bagging-based algorithm where only a subset of features are selected at random to build a forest or collection of decision trees
Gradient Boosting employs gradient descent algorithm to minimize errors in sequential models
Cache awareness and out-of-core computing
Regularization for avoiding overfitting
Tree pruning using depth-first approach
Efficient handling of missing data
XGBoost
Parallelized tree building
In-built cross-validation capability

Figure 6: Evolution of XGBoost algorithm from decision trees (left) and XGB optimizes standard GBM algorithm (right) (*Source: Google.com/XGBoost)*

System Optimization:

Parallelization: XGB approaches use the process of sequential tree building using parallelized implementation. This is becoming a reality because of the interchangeable nature of loops used for building base learners; the outer loop that enumerates the leaf nodes of a tree, and the second inner loop that calculates the features. We can limit parallelization by nesting the loops because without completing the inner loop, the outer loop can't be started. In order to boost run time, interchanging the order of loops using initialization through a global scan of all instances and sorting using parallel threads. This switch improves algorithmic performance by offsetting any parallelization overheads in computation.

Tree Pruning: The criterion that is used for stopping tree splitting within the GBM framework is greedy in nature and depends on the negative loss criterion at the point of the split. XGB has a *max_depth* parameter instead of criterion first and starts pruning trees backward. In order to improve computational performance significantly, it uses a 'depth-first approach.

Hardware Optimization: XGB algorithm has been designed in a way to make efficient use of hardware resources. This is mainly achieved by cache awareness by allocating internal buffers in each thread to store gradient statistics. Further enhancements such as 'out-of-core computing optimize available disk space while handling big data-frames that do not fit into memory.

Algorithmic Enhancements:

Regularization: It penalizes more complex models through both LASSO (L1) and Ridge (L2) regularization to prevent overfitting.

Sparsity Awareness: XGBoost naturally admits sparse features for inputs by automatically 'learning' best missing value depending on training loss and handles different types of sparsity patterns in the data more efficiently.

Weighted Quantile Sketch: XGBoost uses the distributed weighted Quantile Sketch algorithm to find the optimal split points among weighted datasets in an effective manner.

Cross-validation: The algorithm comes with a built-in cross-validation method at each iteration, taking away the need to explicitly program this search and to specify the exact number of boosting iterations required in a single run.

The following process chart (Figure 7) shows the steps that we follow to implement our XGB model in this paper.

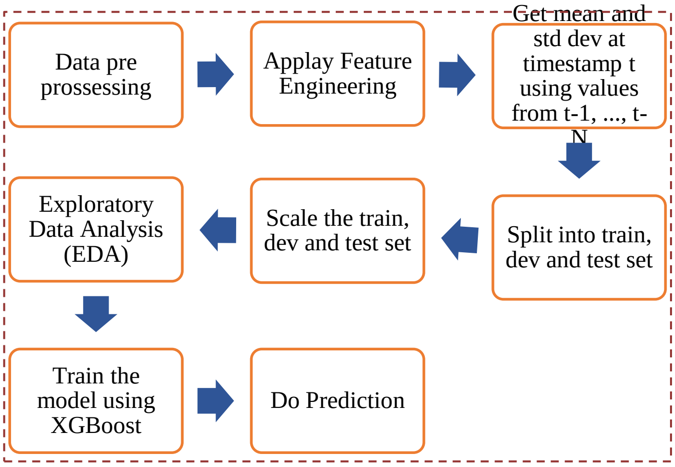

Data pre prossessing
Applay Feature Engineering
Get mean and std dev at timestamp t using values from t-1, ..., t-N
Exploratory Data Analysis (EDA)
Scale the train, dev and test set
Split into train, dev and test set
Train the model using XGBoost
Do Prediction

Figure 7: Steps that taken to implement the XGB model

Functional Requirement

Functional requirements are the ones that enable our users to interact with our system and again our system to interact with other systems. Examples of external interface requirements are user interfaces, hardware interfaces, software interfaces, and communication interfaces. Yet, for Ethiopia Coffee Price Predictor (ECPP) we do not have any hardware interface.

User Interfaces: For the user interface part, we used an android application. The main aim of user interface design is to produce a user interface that makes it easy, efficient, and enjoyable (user-friendly) to do prediction in the way which produces the desired result. This means that in order to get the desired output the operator needs to provide minimal input and the application minimizes undesired outputs to the human.

Android Technology Explained

Android is a software package and also a Linux based operating system for mobile devices such as tablet computers and smartphones. Java language and Java core libraries are used to write the android code even though other languages can be used. They are first compiled to Dalvik executables to run on the Dalvik virtual machine, which is a virtual machine mainly designed for mobile devices. Developers may download the SDK from the Android website. This includes tools, sample code and relevant documents for creating Android apps.

Android first developed by Android Inc., but Google bought in 2005, Android was unveiled in 2007 with the first commercial Android device launched in September 2008. The current stable version is Android 10 introduced on September 3, 2019. The main Android source code is known as Android Open-Source Project (AOSP), which is licensed under the Apache License. This makes it suitable for a variant of Android to be developed on a range of other electronics, such as game consoles, digital cameras, PCs and others, each with a specialized user interface. Some well-known derivatives include Android TV for televisions and Wear OS for wearables, both developed by Google [38].

Android's source code used as the foundation of different ecosystems for examples, Google which is associated with a suite of proprietary software called Google Mobile Services (GMS). GMS frequently comes pre-installed on said devices, this includes Google core apps like Gmail, the digital distribution platform Google Play and associated Google Play Services development platform, and usually apps such as the Google Chrome web browser. So, these apps are licensed by manufacturers of Android devices certified under standards imposed by Google. Android have many competitor ecosystems include Amazon.com's Fire OS, or LineageOS. Distribution of the Software is generally offered through application stores like Google Play Store or Samsung Galaxy Store, or open-source platforms like Aptoide or F-Droid, which use software packages in the APK format.

Android has been the best-selling OS worldwide on smartphones starting from 2011 and on tablets since 2013. In the year 2017 android reaches over two billion monthly active users, which makes It the largest installed base of any operating system, and as of January 2020, the Google Play Store features over 2.9 million apps.

The famous hardware platform for Android is ARM (the ARMv7 and ARMv8-A architectures), with x86 and x86-64 architectures also officially supported in later versions of Android. Android-x86 the unofficial project provided support for x86 architectures ahead of the official support. The ARMv5TE and MIPS32/64 architectures were also supported but removed in later Android releases versions. Starting from 2012 Android devices processors start to appear with Intel, including phones and tablets and it gain support for 64-bit platforms, Android was first made to run on 64-bit x86 and then on ARM64. Since Android 5.0 "Lollipop", 64-bit variants of all platforms are supported in addition to the 32-bit variants [38].

The Minimum Requirement for the amount of RAM for devices running Android 7.1 the ranges start from in practice 2 GB for latest hardware, down to 1 GB for the most common screen, to the least 512 MB for the lowest spec 32-bit smartphone. The suggestion for Android 4.4 is to have at least 512 MB of RAM, while for “low RAM” devices 340 MB is the required minimum amount. Android 4.4 requires a 32-bit ARMv7, MIPS or x86 architecture processor with an OpenGL ES 2.0 compatible graphics processing unit (GPU). Android supports OpenGL ES 1.1, 2.0, 3.0, 3.1 and as of the latest major version, 3.2 and since Android 7.0 Vulkan (and version 1.1 available for some devices). But some applications may explicitly require a certain version of the OpenGL ES, and suitable GPU hardware is required to run such applications [38].

Android project for prediction algorithms

For this paper, we build an android application in order to make it easy for users to do predictions using the application. The reason why we chose android application is that, in Ethiopia as well as around the word mobile is becoming our biggest part of our life's and we use it in everyday activity so we found it reasonable to use a mobile application to do our user interface to make it accessible for most people.

Implementing machine learning algorithms in a mobile application is a bit complicated because of the processing power of the mobile phones some algorithms can even take days to run even if we use laptops or desktops which depends on their processing power.

In order to implement machine learning algorithms on mobile application google provide a solution called TensorFlow Lite. TensorFlow Lite is a set of tools to help developers run TensorFlow models on any mobile device, embedded device, and IoT devices. It enables on-device machine learning inference with low latency and a small binary size [39]. Because TensorFlow Lite (TL) is a new platform it is still in a technological preview state, because of that all TensorFlow features are not currently supported and also going to be the reference for mobile and embedded devices in the future. Because TensorFlow Lite doesn't support all TensorFlow features because of that we got into trouble converting our model completely into TensorFlow Lite because it doesn't support control flow ops such that Enter, Exit, Merge, Switch that we use in our code.

The best solution we come up with is that in order to be able to display prediction results for all the algorithms that we did whether it uses TensorFlow or not is to get the result of the algorithms in CSV format after we finish running the algorithm. Because we are dealing with numerical data that is both our input and output is in numerical form, we make the algorithms to output a CSV file so that we can easily store it in the online database and access the data from our mobile phones. After accessing the data from the online database, we can do whatever displaying method we want using java programming in android applications.

The mobile application will fetch numerical prediction output from the online MySQL database using PHP API (see Figure 8). An API is good for communication between an app and a server. When we send a user request to the online server using an Android server the response from your request is fast in JSON. The RESTful API supports the most commonly used HTTP methods (GET, POST, PUT, and DELETE) and explained as follows: *GET to retrieve and search data, POST to add data, PUT to update data, and DELETE to delete data.*

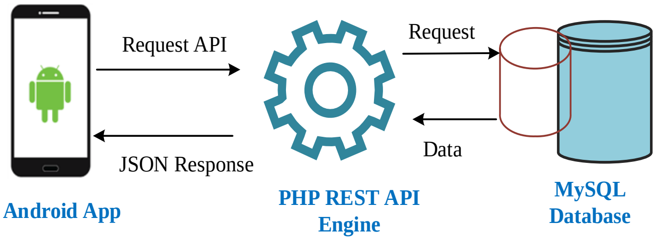


Figure 8: API between the android app and MySQL database

After receiving the response in JSON format we will extract the JSON object and from the JSON object we will extract the elements inside the JSON object after that we will use it to plot the prediction graph and prediction analysis.

The first activity contains a card with a grid view format for the list of coffees that we did prediction for. For this research purpose we choose the top three coffee types so the list contains these coffee types to choose from and the daily price. The second activity for each card contains the prediction algorithms, the user can choose between linear regression, extreme gradient boosting or Long Short-Term Memory (LSTM). In case the user didn't have knowledge about the algorithms the app will put the top performed algorithm prediction graph on the top of the application screen.

The Application is designed to be easy to use for everybody. On the front page, the activity contains the coffee types code (LUBP4, LUBP3, ULK5) which both the farmers and the investors familiar with (see Figure 9). The fourth card in the first activity which is Today Price will contain the updated price of the market every day which the administrator of the app will update every day. Because the app is internet-based it fetches the data from the online server this gives the administrator to update the price every day.

VII. SYSTEM DESIGN AND ANALYSIS

This section contains information on the design decisions that went into the prediction algorithm design and Ethiopian Coffee Price Predictor (ECPP) mobile application. The System Overview is a section to introduce and give a brief overview of the design, this allows the reader and user of this document to orient them to the design and to make them see the summary before proceeding into the details of the design. The architecture patterns are outlined and explained, the system architecture is a way to give the overall view of a system and to place it into context with external systems. The entire system is shown in a use-case and sequential diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

The main point here is whole design concept of the prediction models that we use for this research paper which are XGB, LSTM, and LR, and comparative analysis between them. The whole design concept of ECPP is based on human-centered computing with a user-friendly design that can be accessed using an Android mobile application and displaying the prediction model.

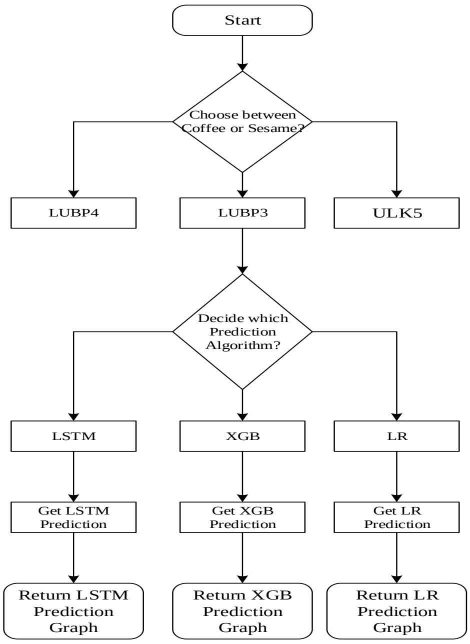

Start
Choose between Coffee or Sesame?
LUBP4
LUBP3
ULK5
Decide which Prediction Algorithm?
LSTM
XGB
LR
Get LSTM Prediction
Get XGB Prediction
Get LR Prediction
Return LSTM Prediction Graph
Return XGB Prediction Graph
Return LR Prediction Graph

Figure 9: Request processing in the back-end

System Features

The major features of the Price Predictor will be the following:

- *Prediction using the LSTM model*: we use one of the recurrent neural network methods which are LSTM to predict the coffee price in the Ethiopian market.
- *Prediction using the XGB model*: we used the advanced version of the Gradient Boosting model which is Extreme Gradient Boosting to predict the coffee price in the Ethiopian market.
- *Prediction using the LR model:* we use one of regression models to predict the price of Ethiopia coffee which is Linear Regression.
- *Storing prediction output* we will store all of our model's output in the MySQL database and we will access It on our mobile phone.
- *Android Application:* In order to display our prediction result, we designed ECPP and fetch the prediction results from the database to display.
- *User interface API:* we used a PHP based API to request data from MySQL database and we

will receive the replay in JSON format in order to build the ECPP.

Technologies Used

The prediction models will be developed in python. The user interface will be developed in java and XML. The API is developed using PHP, we will create tables in MySQL. The system uses external libraries that it relies on:

- *Tensor Flow:* TensorFlow is an open-source library for numerical computation that makes machine learning faster and easier Created by the Google Brain team, TensorFlow is a library for numerical computation and large-scale machine learning it also bundles together a slew of machine learning and deep learning models and algorithms and makes them useful by way of a common metaphor. Tensor Flow mainly uses Python to provide a convenient front-end API for building applications with the framework, while executing those applications in high-performance C++.
- *Scikit-learn*: It is a machine learning library for the Python programming language [7]. It supports various types of classification, regression and also clustering algorithms including random forests, support vector machines, k-means, gradient boosting, and is designed to incorporate with the Python libraries NumPy as well as SciPy.
- *Matplotlib:* It is python libraries that provide functions to plot various data sets. It uses NumPy to handle large arrays of data sets. With the current organization, we created the tool to plot various complex data sets by using this library. It includes features like zoom in, action forward-backward, drag is so impressive.
- *Seaborn*: It is a Python data visualization library based on matplotlib. It contains a high-level and easy to use interface for drawing attractive and informative statistical graphics [40].
- *XGBoost*: It is a gradient boosting library designed to be highly efficient, flexible and portable. It used to implement machine learning algorithms under the Gradient Boosting framework. XGB provides a parallel tree boosting also known as GBDT, GBM that solves many data science problems in a fast and accurate way.
- *Pandas:* It is built on top of the Python programming language. It is fast, powerful, flexible and easy to use open-source data analysis and manipulation tools.
- *NumPy:* It is a Python library that provides a multidimensional array object and various derived objects such as masked arrays and matrices, and an assortment of routines for fast operations on arrays [41].
- *Keras:* Keras is written in Python programing language and it is a high-level neural networks API, and capable of running on top of TensorFlow, Theano or CNTK and it was developed with a focus on allowing fast experimentation. In order to be able to go from idea to actual result with the small delay is key to doing good research [42].
- *datetime:* The module supplies classes for manipulating dates as well as times in both simple and complex ways. The date and time arithmetic are supported and the main focus of the implementation is extracting the attribute output, formatting and manipulating in an efficient manner [43].

VIII.RESEARCH DESIGN AND METHODOLOGY

The methodology is a formalized approach for implementing the Systems Development Life Cycle (SDLC). There are various methodologies being used in the IT area these days such as waterfall, Rapid, scrum and to be specific it is a waterfall. To generate the Object-Oriented Analysis and Design (OOAD) we have.

Model performance evaluation methods (Loss Functions)

In order to do comparative analysis its necessary to measure the model performance using different methods so for our case to measure the performance of our three algorithms which are LR, XGB and LSTM we used the following three performance evaluation methods which are:

1. RMSE (Root Mean Square Error)
2. R-squared ($\boldsymbol{R}^2$)
3. MAPE (Mean Absolute Percent Error)

Root Mean Square Error (RMSE): *The* Root Mean Square Error (RMSE) is the standard deviation of the residuals or prediction errors. Residuals used to measure how far from the regression line data points are. RMSE is a measure of how to spread out these residuals are from the prediction line or in other words it tells you how concentrated the data is around the line of best fit. Root mean square error is commonly used in climatology, forecasting, and regression analysis to verify experimental results but for our case, we focus on the regression part.

RMSE can be easily interpreted compared to MSE because RMSE units match the unit of the output. RMSE can be negative or positive as the predicted value under or overestimates the actual value. Squaring the residuals, averaging the squares, and taking the square root gives us the RMSE. You then use the RMSE as a measure of the spread of the y values about the predicted y value and represented by the formula in equation [1].

$$RMS\ Errors = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}} \qquad [1]$$

$\boldsymbol{R}^2$: R-squared ($\boldsymbol{R}^2$) measures the proportion of the variance for a dependent variable that's explained by an independent variable and their correlation explains the strength of the relationship between an independent and dependent variable in a good manner, R-squared explains to what extent the variance of one variable explains the variance of the second variable. So, if the R2 of a model is 0.50, then approximately half of the observed variation can be explained by the model's inputs and again if $\boldsymbol{R}^2$ of a model is 1, then observed variation can be fully explained by the model's inputs. To simply put it R-squared is the percentage of variation explained by the relationship between two variables which is the dependent and independent variable and represented by the formula in equation [2].

R-squared values range from 0 to 1 and are commonly expressed as percentages from 0% to 100%. An R-squared of 100% means that all dependent variable are completely explained independent variable(s) you are interested in [37].

$$R^2 = 1 - \frac{Explained\ variation}{Total\ variation} \qquad [2]$$

Mean Absolute Percent Error (MAPE): In Order to understand RMSE first, we need to see MAE. MAE is obtained by calculating the absolute difference between the model prediction and the true (Actual) value. MAPE the equivalent to MAE but provide the error in percentage form and overcome MAE limitations and it is represented by the equation [3]. And also, MAPE might exhibit some limitations if the data point value is Zero.

$$APE = \frac{100}{n}\sum_{t=1}^{n}\frac{|A_t - F_t|}{A_t}\ \% \qquad [3]$$

Figure 10 shows a high-level use case diagram explain the sub-system modules we have used for our design. We have presented our system operation in a matter of A sequence diagram (Figure 11), which is an interaction diagram. From the name, it is clear that the diagram deals with some sequences, which are the sequence of messages flowing from one module to another. The relationship between the components of a system is very important from the implementation and execution perspective. So, the Sequence diagram is used to visualize the sequence of calls in a system to perform specific functionality.

When we see the general picture of the system operation in our paper first, we have to see the prediction algorithms. Our main and important objective in this paper is to perform a Comparative analysis of the algorithms that we used and also choose the best-performed algorithm in order to use it for further applications. After we perform comparative analysis and choose the best performing algorithm based on the loss functions that we used, we will see how to implement this algorithm in android mobile applications.

Figure 10 shows a high-level use case diagram explain the sub-system modules we have used for our design. We have presented our system operation in a matter of A sequence diagram (Figure 11), which is an interaction diagram. From the name, it is clear that the diagram deals with some sequences, which are the sequence of messages flowing from one module to another. The relationship between the components of a system is very important from the implementation and execution perspective. So, the Sequence diagram is used to visualize the sequence of calls in a system to perform specific functionality.

When we see the general picture of the system operation in our paper first, we have to see the prediction algorithms. Our main and important objective in this paper is to perform a Comparative analysis of the algorithms that we used and also choose the best-performed algorithm in order to use it for further applications. After we perform comparative analysis and choose the best performing algorithm based on the loss functions that we used, we will see how to implement this algorithm in android mobile applications.

In order to use the system app first, the user needs to open the app and choose the coffee type or sesame to get the prediction of the specific item that we want to see. When we choose the type of the item then our mobile will send a request to the database to fetch the required data and the database will take the data from the predictive algorithm output. After that, we will be able to display it in our app screen in the form of a graph which will show all the previous prediction result too. Figure 11 shows the sequence diagram of the interaction between all the modules.

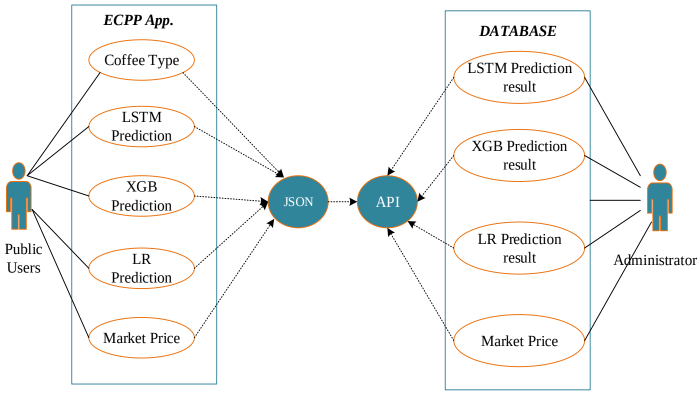


Figure 10: High-level Use case diagram of ECPP

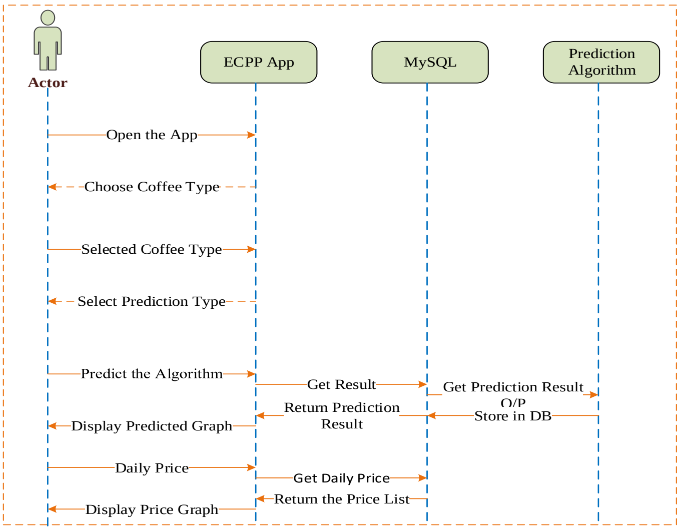


Figure 11: Sequence diagram for ECPP

IX. IMPLEMENTATION, TESTING, AND ANALYSIS

This section has three main parts, the implementation part covers the discussion of the execution proposed solution and describes the formal testing to be performed and again we will perform the comparative analysis between the algorithms. Finally, we will see the result implementation of the algorithm in the mobile application.

Data Description

The dataset that we used to test our algorithm is obtained from Ethiopian Commodity Exchange (ECX) which include coffees price dataset from 2012 to 2018 and sesame dataset from 2012 to 2019, each dataset has 11 attributes: Date, Symbol, Wearhouse number, Production Year, open, Close, High, Low, Change, Percentage Change, Volume. The dataset was obtained at https://data.mendeley.com/datasets/c55sbp7dgv/1

1. Coffee (LUBP4): which have 1540 instances and 11 attributes
2. Sesame: which have 7205 instances and 11 attributes

We also used Sesame dataset from 2012 to the end of 2019 which have 7205 instances and 11 attributes. The reason why we choose this different dataset is that economically *Coffee and Sesame* play a great roll in the *Ethiopian Economy* by being the largest export item. Apart from being the largest export item we also need different kinds of datasets in order to make compression according to how big the data set is. As we mentioned above Sesame dataset has 7205 instances and the coffee data set has a maximum of 1540 instances so using these two, we can be able to evaluate the effect of the data set on the algorithms.

Data is obtained in Excel format and several simple functions are required to be written in order to achieve usable form from raw input data. And panda library is imported the data frame is converted to a $NumPy$ array, also the number seed.

Data Pre-processing

Before we start doing the prediction of Ethiopian coffee market price using three different algorithms, we need to do data pre-processing and data visualization because the more we understood the behaviour of our dataset and the relationship between the attributes the better it will be for us to use it in the prediction algorithms and be able to toon the parameter of the algorithms.

In this paper, we implement the following steps in order to ensure the quality of our dataset: *data cleaning, missing values imputation, feature selection, and Exploratory Data Analysis (EDA).* The following plot shows the visualization of the overall data set, we choose to use the adj_close attribute because this is the most important feature and contain more information than the other attributes. Let us define the set of features that we utilized during the implementation: $Adj.Close$: this is an important source of information as this decides the market opening price for the next day and volume expectancy for the day, $Volume$: this is a very important decision parameter as the volume traded has the most direct impact on future stock price than any other feature, and in addition, we used $Feature\ Engineering$ to generate new features because Feature engineering is a creative process and is one of the most important parts of any machine learning project.

Apart from visualizing the data set we also try to see its relationship with the volume of the coffee because the price of the coffee directly affected by the volume or the amount of the coffee that is sold. When we see the relationship between the adj_close value and the Volume of the coffee we can clearly see that when the volume or the supply is high the price of the coffee will be law, which is logical because in the economics principle when the supply is high the price of the item will be law. For example, between 2012 and 2013 the volume reaches the highest but the adj_close price is the lowest of all so when the supply is the price will be low.

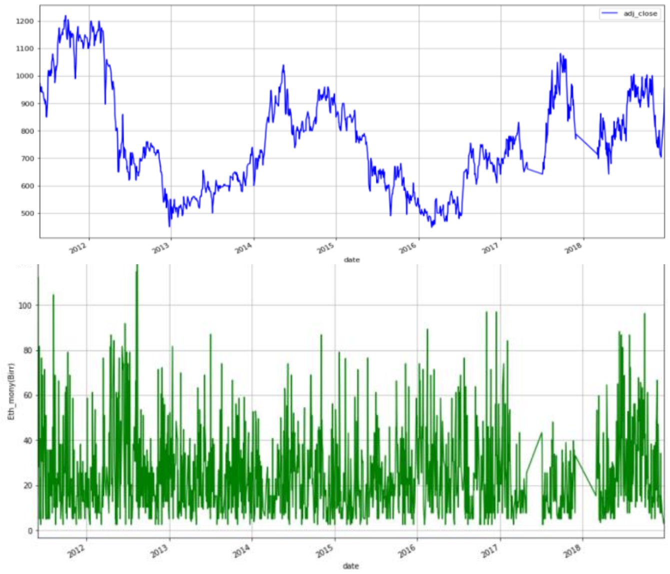


Figure 5. 1: The adj_close and volume relationship of coffee dataset

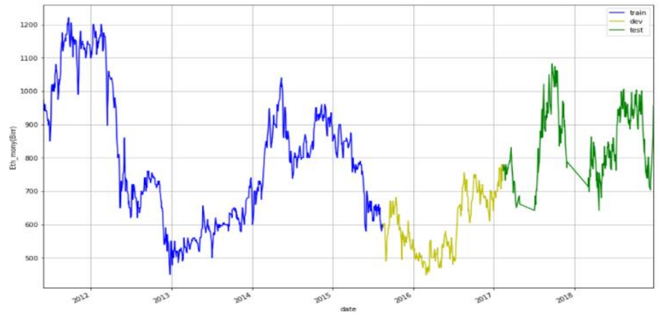


Figure 5.2: This plot show how we split the data set in to training, validation and testing set.

The next plot shows average price for each month which help us to visualize the behaviour of the market on each month (Figure 5.3-top). The next plot (Figure 5.3-bottom) computes the average price for each day of month, again this helps us to visualize the behaviour of the market on each day of month.

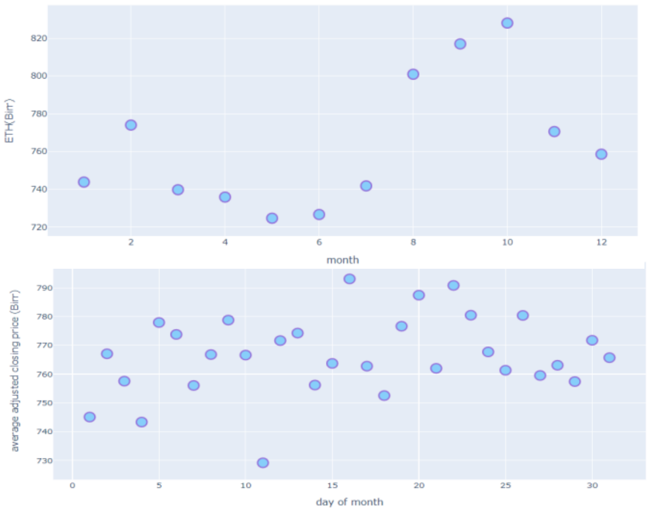


Figure 5.3: Average price for each month (top) and average price for each day of month (bottom)

A: Linear regression prediction result

Performing linear regression on sesame dataset

The Sesame data set have 7205 instances and 11 attributes (From the year 2012 to the end of 2019) which we believe is good data set when it comes to training a machine learning algorithm. The source of the data set is the Ethiopian commodity Exchange (ECX). When we visualize the data using Explanatory data analysis we can see as follows (Figure 5.4) which shows training, validation and test dataset which looks like the following:

- $num\_train = 3864$
- $num\_cv = 1287$
- $num\_test = 1287$
- $train.shape = (3864, 12)$
- $cv.shape = (1287, 12)$
- $train\_cv.shape = (5151, 12)$
- $test.shape = (1287, 12)$

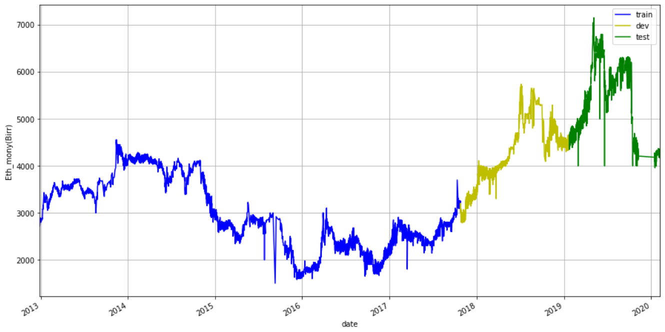


Figure 5.4: Sesame EDA

The next step will be applying the linear regression algorithm by using $sklearn\ library$. After applying the linear regression model on our dataset, we can measure the loss or the accuracy of the model by using three different methods which are MAPE, RMSE, and R-squared.

From the Figure 5.5a, b and c of RMSE, MAPE and R-squared we can understand that for RMSE and MAPE we get optimal value at N=1 and N=15. Also, for the R-squared, we get the optimum value at N=1 and N=15. Using this information, we can understand how the actual dataset to fit the regression line and we will explain it at the end of this experiment.

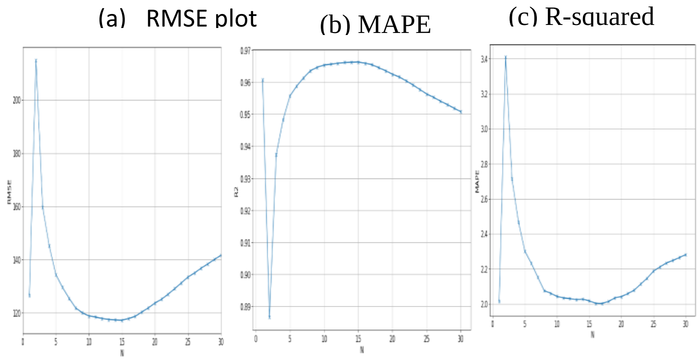
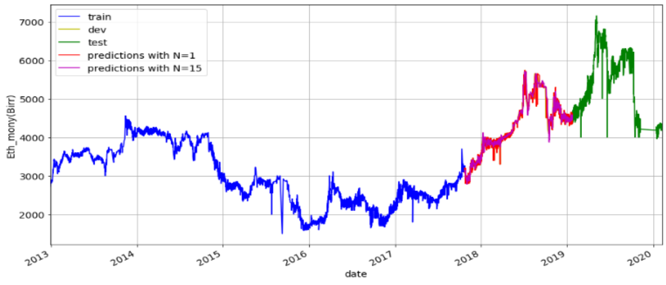

(a) RMSE plot (b) MAPE (c) R-squared

Figure 5.5: The plots for RMSE, MAPE and R-squared on sesame dataset using LR algorithm

The next step will be plotting the prediction on the development set before we plot it on the test set. The prediction on the development set shows the relationship between the two optimal points which is at N=1 and N=15 (N is the number of samples that we use to predict the next value) as we have seen on MAPE, RMSE, and R-squared. The Figure 5.6 shows the whole dataset plot in order to see the clear visualization the second plot will show the zoom-in perspective.



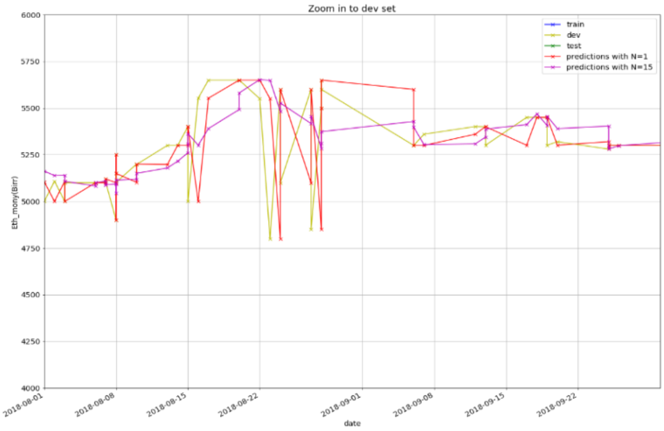


Figure 5.6: Prediction on development set of sesame dataset using LR

The final Step for this algorithm is to do the final prediction on the test set. This means we are going to apply our linear regression on our test set. The following two plots will show our prediction on the training set, the first plot will help to visualize the whole testing set prediction but because of the large data set it hard to identify the prediction plot so in the second plot we will show the zoom-in plot in order to make the prediction more visible.

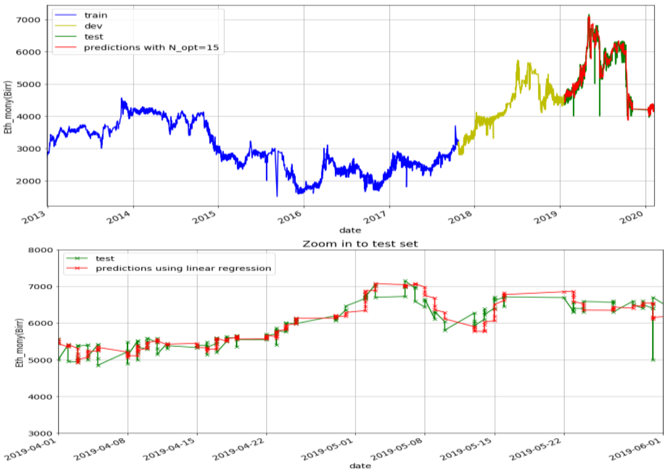


Figure 5.7: Prediction on the test set of sesame dataset using LR

Generally, when we did the prediction using Linear Regression, we get the results that we showed previously in the plot (see Figure 5.6) but we can only visualize how well the prediction line fits the real dataset. But in order to evaluate how the model perform we need to perform model evaluation methods, this is the loss functions that we used, which are RMSE, MAPE and R-squared. The result that we get from the algorithm is as follow:

- $RMSE = 169.275$
- $R2 = 0.955$
- $MAPE = 2.062\%$

Consequently, the question here is that what does this loss functions number means? When we see the first one which is RMSE=169.275, it measures of how-to spread-out residuals are from the prediction line or in other words it tells you how concentrated the data is around the line of best fit and also RMSE units match the unit of the output which in this case means the residuals are spread out from the prediction line by 169.275 Ethiopian money (Birr). The unit is becoming like this because as I mentioned above RMSE units match the unit of the output so according to this the result is good.

The Second method that we used to measure the accuracy is R^2 which is equal to 0.955. this means 95.5 % of the variation in closing price is accounted for by its regression on date or the closing/date relationship accounts for 95.5% of the variation, this means that hardly any of the variations in the data is explained by the close/date relationship which is very good news.

The last one is MAPE which is equal to 2.062%. this value is obtained by calculating the percentage of the absolute difference between the model prediction and the true (Actual) value. Therefore, based on this result our model performs well with only 2.062% error.

Performing linear regression on Coffee dataset

The Coffee data set have 1540 instances and 11 attributes (From the year 2012 to 2018) and the source of the data set is the Ethiopian commodity Exchange (ECX).

The way we are going to use linear regression here is that we will fit a linear regression model to the previous N values, and use this model to predict the value on the current day. The below plot is an example of $N = 11$. The actual adjusted closing prices are shown as dark blue cross, and we want to predict the value on day 12(yellow square). We will fit a linear regression line (light blue line) through the first 11 actual values, and use it to do the prediction on day 12 (light blue circle).

When we visualize the data using Explanatory data analysis we can see as follow which shows training, validation and test data set which looks like the following:

- $num\_train = 925$
- $num\_cv = 307$
- $num\_test = 307$
- $train.shape = (925, 13)$
- $cv.shape = (307, 13)$
- $train\_cv.shape = (1232, 13)$
- $test.shape = (307, 13)$

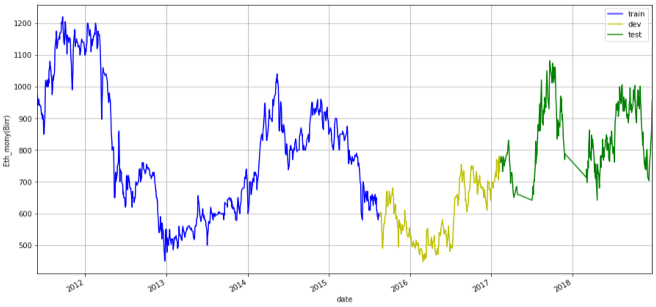


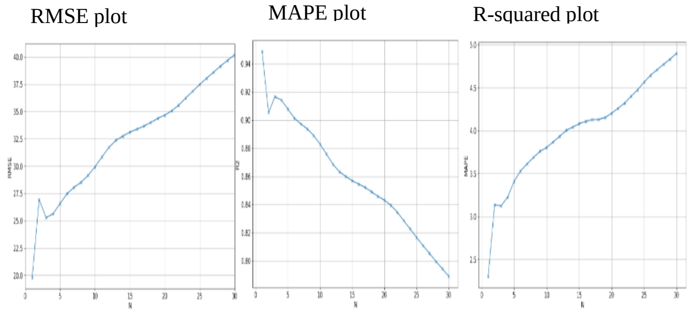
Figure 5.8: Coffee EDA

Setting the N value can be adjusted with different numbers; we took this small value to make it more visible on the graph. For example, if we take $N = 5$ and if we want to predict the price on the 6th day because the algorithm predicts the next day value by evaluating N days from the past result. For the prediction purpose, the prices of five previous days are very much useful.

The next step will be applying the linear regression algorithm by using $sklearn\ library$. After applying the linear regression model on our data set, we can measure the loss or the accuracy of the model by using three different methods which are MAPE, RMSE, and R-squared.

From Figure 5.9 of RMSE (a), MAPE (b), and R-squared (c), we can understand that for RMSE and MAPE we get optimal value at $N = 1\ and\ N = 3$. Also, for the R-squared, we get the optimum value at $N = 1\ and\ N = 3$. Using this information, we can understand how the actual data set to fit the regression line and we will explain it at the end of this experiment.

The below plot shows the RMSE between the actual and predicted values on the validation set, for various values of N. We will use N=3 because it gives the lowest RMSE. For this algorithm, the lowest RMSE achieved at $N = 1\ followed\ by\ N = 3$.


RMSE plot

MAPE plot

R-squared plot

Figure 5.9: The plot for RMSE, MAPE and R-squared using coffee dataset

The next step will be plotting the prediction on the development set before we plot it on the test set. The prediction on the development set shows the relationship between the optimal points which is at $N = 1\ and\ N = 3$ (N is the number of samples that we use to predict the next value) as we have seen on MAPE, RMSE, and R-squared. Figure 5.10-top shows the whole dataset in the plot in order to see the clear Figure 5.10-bottom the second plot will show the zoom-in perspective.

When it comes to the prediction on the test set, we can observe in the following graph that the prediction is not that good, which is having a problem in handling high and low changes.

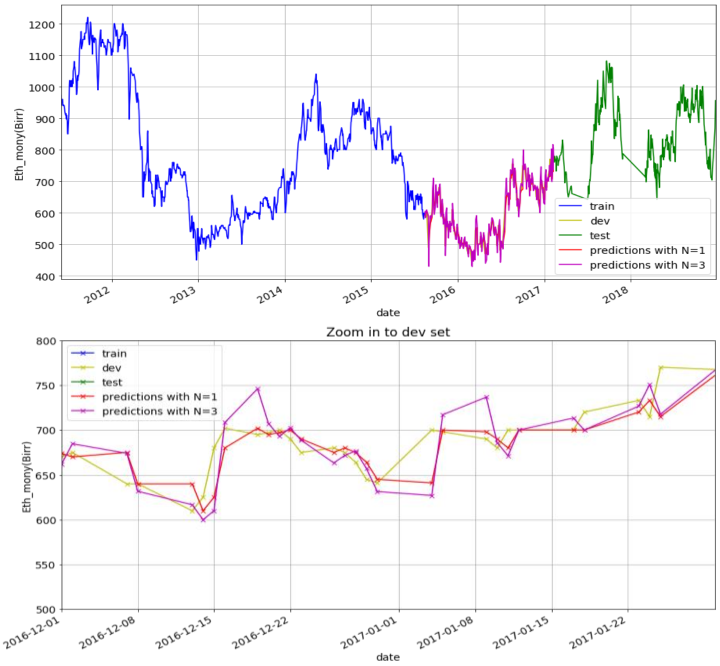
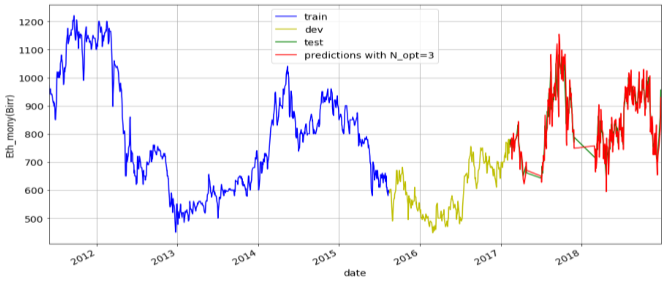


Figure 5.10: The prediction on development set

The final step for this algorithm is to do the final prediction on the test set. This means we are going to apply our Linear regression on our test set. The following two plots (Figure 5.11) shows our prediction on the test set, the first plot will help to visualize the whole testing set prediction but because of the large dataset it hard to identify the prediction plot so in the second plot we will show the zoom-in plot in order to make the prediction more visible.



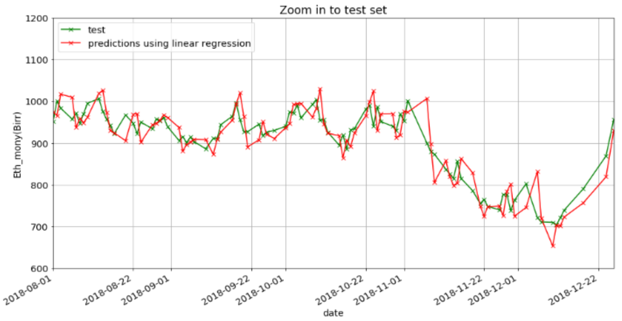


Figure 5.11: Prediction on the test set

Generally, when we did prediction using Linear Regression, we get the results that we showed previously but we can only visualize how well the prediction line fits the real data set. But in order to evaluate how the model perform we need to perform model evaluation methods that are the loss functions that we used which are $RMSE, MAPE, and\ R-squared$. The result that we get from the algorithm is as follow:

- $RMSE\ =\ 42.316$
- $R2\ =\ 0.836$
- $MAPE\ =\ 3.738\%$

When we see the first one which is RMSE=42.316, it measures of how-to spread-out residuals are from the prediction line or in other words it tells you how concentrated the data is around the line of best fit and also RMSE units match the unit of the output which in this case means the residuals are spread out from the prediction line by 42.316 Ethiopian money (Birr). The unit is becoming like this because as we mentioned above RMSE units match the unit of the output so according to this the result is good.

The Second method that we used to measure the accuracy is R^2 which is equal to 0.836. this means 83.6 % of the variation in closing price is accounted for by its regression on date or the closing/date relationship accounts for 83.6% of the variation, this means that hardly any of the variations in the data is explained by the close/date relationship which is very good news.

The last one is MAPE which is equal to 3.738%. This value is obtained by calculating the percentage of the absolute difference between the model prediction and the true (Actual) value. Consequently, based on this result our model performs well with only a 3.738% error.

Comparative analysis on applying linear regression on Sesame and Coffee dataset

In the previous experimental results, we showed the result of applying Linear regression on the Sesame dataset and coffee dataset. In this section, we will do a comparative analysis between these two results. Our main objective of this compression is to see the effect of the dataset on the Linear Regression algorism and prediction.

TABLE 5.1: Comparative analysis on applying linear regression on Sesame and Coffee dataset

No.	Coffee data set (1540 instances and 11 attribute)	Sesame data set (7200 instances and 11 attributes)	Comparative analysis
1.	RMSE = 42.316	RMSE = 169.275	Here we cannot really say this data set is better but when the data set is big the RMSE also have high probability of increasing and we should not forget that the unit is Birr (which is Ethiopian money). This just measures of how spread-out residuals are from the prediction line in terms of Birr.
2.	R2 = 0.836	R2 = 0.955	When we see the R-squared the Sesame

			data set perform better than the coffee data set by scoring 95.5% over 83.6%
3.	MAPE = 3.738%	MAPE = 2.062%	Again, for Mean Absolute Present Error Sesame data set perform better by scoring 2.062% error while the coffee data set perform with 3.738% error

Extreme Gradient Boosting

Performing XGB on coffee dataset

The name XGBoost refers to the engineering goal to push the limit of computational resources for boosted tree algorithms. Ever since its introduction in 2014, XGBoost has proven to be a very powerful machine learning technique and is usually the go-to algorithm in many Machine Learning competitions.

The first step in developing this algorithm is loading the coffee data set using the panda library which has 1540 instances and 11 attributes (From 2012 to 2018). After we finish loading the dataset, we will perform feature engineering on the data set. We will train the XGBoost model on the train set, tune its hyperparameters using the validation set, and finally apply the XGBoost model on the test set and report the results. The features to use are the adjusted closing prices of the last N days, as well as the volume of the last N days. In addition to these features, we can do some feature engineering.

- *Mean 'adj_close' of each month*
- *Difference between high and low of each da*
- *Difference between open and close of each d*
- *Mean volume of each month*

When we visualize the data using explanatory data analysis, which shows how we split training, validation and test data set which looks as the following:

- $num\_train = 922$
- $num\_cv = 307$
- $num\_test = 307$
- $train.shape = (922, 27)$
- $cv.shape = (307, 27)$
- $train\_cv.shape = (1229, 27)$
- $test.shape = (307, 27)$

After we split the dataset, we will scale the train, development and set test but first, we only scale the training dataset, and not the entire dataset to prevent information leak, then we will convert the $NumPy$ array back into a panda data frame. Then we will scale the train and development set and we will convert the $NumPy$ array back into a panda data frame, then we will scale development set at this point we can say the entire development set is scaled (see Figure 5.16). Finally, we perform scaling on the test set.

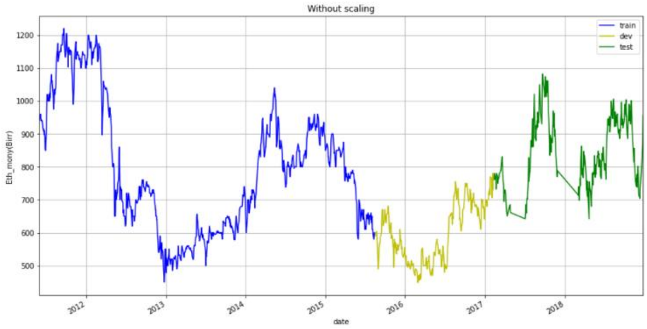


Figure 5.16: EDA for coffee dataset

At this point we can train the model using XGB and perform prediction on the train, development and we perform prediction on the test set after we done parameter tuning.

TABLE 5.3: Prediction loss function on train, development and test set

Loss functions	Prediction on train set	Prediction on dev. set	Prediction on test set
RMSE	16.204	19.916	34.748
MAPE	1.470%	2.36%	3.194%

Generally, when we did prediction using XGB, we get the results that we showed previously but we can only visualize how well the prediction line fits the real dataset (Table 5.3). But in order to evaluate how the model perform we need to perform model evaluation methods those are the loss functions that we used which are RMSE, MAPE and R-squared. The result that we get from the algorithm is as follow:

- $RMSE\ on\ test\ set\ =\ 34.748$
- $MAPE\ on\ test\ set\ =\ 3.194\%$
- $R-squared\ =\ 0.946$

Let see loss functions numbers in this paragraph, when we see the first one which is RMSE=34.748, it measures of how-to spread-out residuals are from the prediction line or in other words it tells you how concentrated the data is around the line of best fit and also RMSE units match the unit of the output which in this case means the residuals are spread out from the prediction line by 34.748 Ethiopian money (Birr). The unit is becoming like this because as I mentioned above RMSE units match the unit of the output so according to this the result is better from the Linear regression result.

The second method that we used to measure the accuracy is MAPE which is equal to 3.194%. This value is obtained by calculating the percentage of the absolute difference between the model prediction and the true (Actual) value. Therefore, based on this result our model performs well with only 3.194% error or 96.8% accuracy.

The third method that we used to measure the accuracy is R^2 which is equal $to\ =\ 0.946$. This means 94.6% of the variation in closing price is accounted for by its regression on date or the closing/date relationship accounts for 94.6% of the variation, this means that hardly any of the variations in the data is explained by the close/date relationship which is very good news. The final prediction on the test set graphically will look like the following. the first plot shows the prediction plot (Figure 5.17-top) in the whole test set and the second plot show the zoom-in view of the plot in order to make it more visible (Figure 5.17-bottom).

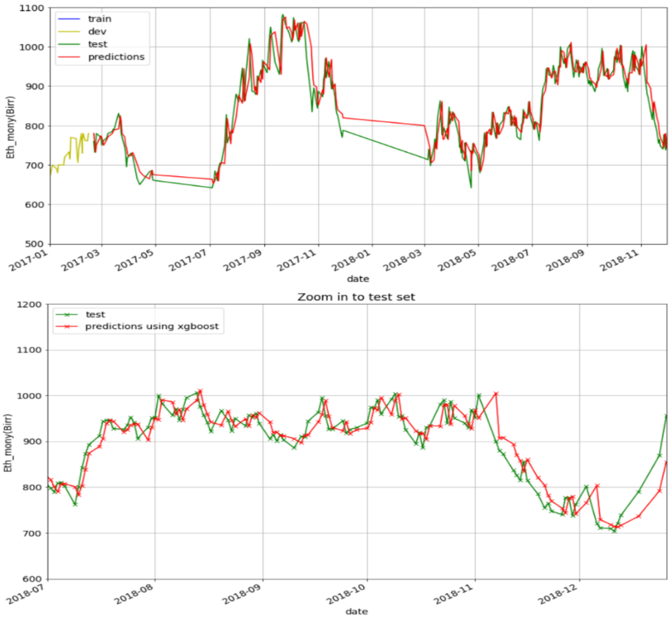


Figure 5.17: Final result on applying XGB on coffee dataset

Performing XGB on sesame dataset

The first step in developing this algorithm is loading the coffee data set using the panda library which has 7205 instances and 11 attributes (From 2012 to 2019). After we finish loading the dataset, we will perform feature engineering on the dataset. We trained the XGB model on the train set, tune its hyperparameters using the validation set, and finally apply the XGB model on the test set and report the results. The features to use are the adjusted closing prices of the last *N* days, as well as the volume of the last *N* days. In addition to these features, we can do some feature engineering.

- $Mean\ 'adj\_close'\ of\ each\ month$
- $Difference\ between\ high\ and\ low\ of\ each\ day$
- $Difference\ between\ open\ and\ close\ of\ each\ day$
- $Mean\ volume\ of\ each\ month$

When we visualize the data using explanatory data analysis, which shows how we split training, validation and test data set which looks as the following:

- $num\_train \ = \ 3852$
- $num\_cv \ = \ 1283$
- $num\_test \ = \ 1283$
- $train.shape \ = \ (3852, 100)$
- $cv.shape \ = \ (1283, 100)$
- $train\_cv.shape \ = \ (5135, 100)$
- $test.shape \ = \ (1283, 100)$

After we split the dataset, we will scale the train, development and set test but first, we only scale the training dataset, and not the entire dataset to prevent information leak, then we converted the NumPy array back into a panda data frame. Then we will scale the train and development set and we converted the $NumPy$ array back into a panda data frame, then we will scale development set at this point we can say the entire development set is scaled (see Figure 5.18). Finally, we perform scaling on the test set.

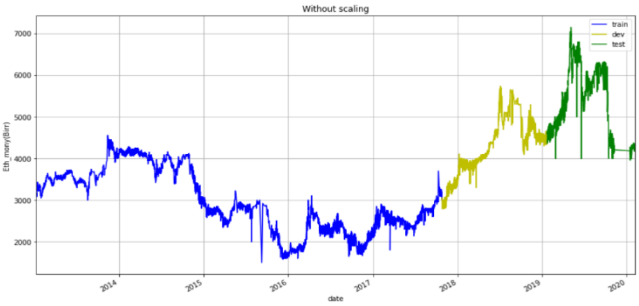


Figure 5.18: EDA for XGB on sesame dataset

At this point we can train the model using XGB and perform prediction on the train, development and we perform prediction on the test set after we done parameter tuning. The result before and after tuning was reengineered.

Usually, when we did prediction using XGB, we get the results that we showed previously but we can only visualize how well the prediction line fits the real dataset. But in order to evaluate how the model perform we need to perform model evaluation methods those are the loss functions that we used which are RMSE, MAPE and R-squared. The result that we get from the algorithm is as follow:

- $RMSE \ on \ test \ set \ = \ 157.225$
- $MAPE \ on \ test \ set \ = \ 1.786\%$
- $R - squared \ = \ 0.945$

Let see loss functions numbers in this paragraph, when we see the first one which is $RMSE = 157.225$, it measures of how-to spread-out residuals are from the prediction line or in other words it tells you how concentrated the data is around the line of best fit and also RMSE units match the unit of the output which in this case means the residuals are spread out from the prediction line by 157.225 Ethiopian money (Birr). The unit is becoming like this because as I mentioned above RMSE units match the unit of the output so according to this the result is better from the Linear regression result.

The Second method that we used to measure the accuracy is MAPE which is equal to 1.786%. This value is obtained by calculating the percentage of the absolute difference between the model prediction and the true (Actual) value. So, based on this result our model performs well with only 1.786% error or 98.2% accuracy.

The third method that we used to measure the accuracy is R^2 which is equal to = 0.945. this means 94.5% of the variation in closing price is accounted for by its regression on date or the closing/date relationship accounts for 94.5% of the variation, this means that hardly any of the variations in the data is explained by the close/date relationship which is very good news. The final prediction on the test set graphically will look like the following (Figure 5.19). the first plot shows the prediction plot in the whole test set (Figure 5.19-top) and the second plot show the zoom-in view of the plot (Figure 5.19-bottom) in order to make it more visible.

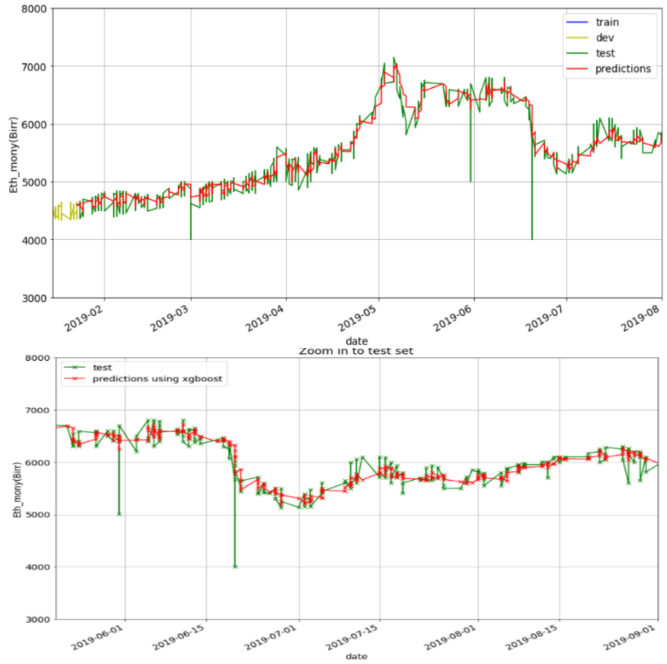


Figure 5.19: XGB prediction plot on the sesame data test set

TABLE 5.4: Prediction loss function on train, development and test set

Loss functions	Prediction on train set	Prediction on dev. set	Prediction on test set
RMSE	71.947	115.625	157.225
MAPE	2.005%	1.900%	1.786%

Comparative analysis between applying XGB on sesame and coffee dataset

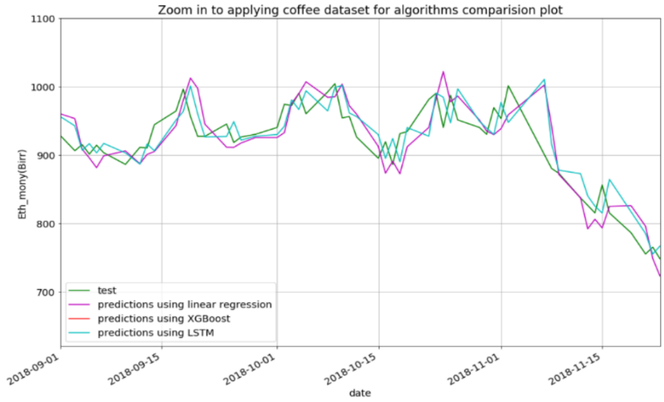
In the previous experimental results, we showed the result of applying LSTM on the sesame dataset and coffee dataset. In this section, we will do a comparative analysis between these two results (Table 5.5). Our main objective of this compression is to see the effect of the data set on the LSTM algorism and prediction.

TABLE 5.5: Comparative analysis between applying XGB on sesame and coffee dataset

No.	XGB on Coffee test set (1540 instances and 11 attribute)	XGB on Sesame test set (7200 instances and 11 attributes)	Comparative analysis between applying LSTM on sesame and coffee data set
1.	RMSE = 34.748	RMSE = 157.226	Here we cannot really say this data set is better but when the data set is big the RMSE also have high probability of increasing and we should not forget that the unit is Birr (which is Ethiopian money). This just measures of how spread-out residuals are from the prediction line in terms of Birr.
2.	R2 = 0.946	R2 = 0.970	When we see the R-squared the Sesame data set perform better than the coffee data set by scoring 97.0% over 97.0%. R-squared results preferable when they approach to 1
3.	MAPE = 3.194%	MAPE = 1.786%	Again, for Mean Absolute Present Error Sesame data set perform better by scoring 1.786% error while the coffee data set perform with 3.194% error

Comparative analysis between prediction algorithms

Largely, the performance of the three models displayed in comparison in the following graph. LSTM is better performed than the other three models that we use in terms of the error that we got and precision to the real value and linear regression does not perform well compare to other models as you can see below (Figure 5.20).



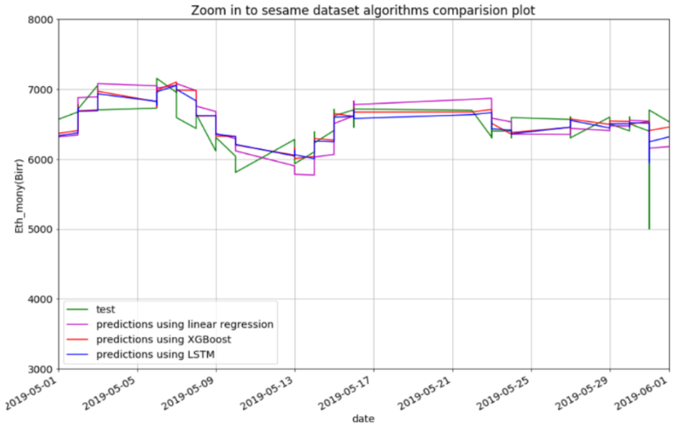


TABLE 5.6: Comparative analysis on applying the three different algorithms on coffee dataset

Loss functions	LSTM on coffee test set	XGB on coffee test set	LR on coffee test set	Comparative analysis
RMSE	34.123	34.748	42.316	When we compare these three algorithms LSTM gives the lowest RMSE value
MAPE	3.176%	3.194%	3.738%	When we see MAPE again LSTM gives the lowest value, but XGB also gives the lowest value only very small difference which is 3.176% to 3.194%
R-Squared	0.897	0.946	0.836	Here the only information we can get is the relationship between the features that we used. this means that hardly any of the variation in the data is explained by the close/date relationship which is good news

Table 5.7: Comparative analysis on applying the three different algorithms on sesame dataset

Loss functions	LSTM on sesame test set	XGB on sesame test set	LR on sesame test set	Comparative analysis
RMSE	153.753	157.225	169.275	When we compare the result of these three algorithms

				LSTM gives the lowest RMSE value but XGB also gives approximate result as LSTM.
MAPE	1.762%	1.786%	2.062%	When we see MAPE again LSTM gives the lowest value, but XGB also gives the lowest value only very small difference which is 1.762% to 1.786%
R-Squared	0.96	0.97	0.96	Here the only information we can get is the relationship between the features that we used. this means that hardly any of the variation in the data is explained by the close/date relationship which is good news

The reason why we choose to use two different data sets (which are coffee and sesame) is that we want to see the effect of using a data set with a different number of instances (sesame has 7205 instances and coffee has 1540). As we can observe on the above comparison tables sesame data set perform better on the algorithms that we use by scoring the lowest MAPE on all algorithms compare to the Coffee data set. So, we conclude that using a data set with more instances and features is better to compare to the one with a low number of instances. When we see the average value of MAPE for sesame is 1.87% and when we see the average value of MAPE for coffee is 3.37%. Consequently, at this point, we can clearly see the bigger the dataset the better to train the algorithms. But here we need to acknowledge that we cannot use RMSE to compare the effect of the two data set on the algorithm that we use because when the dataset to become large so do the RMSE.

Therefore, from the above comparative analysis (Table 5.6 and 5.7), we notice that LSTM performs better than the other two but with a very small difference with XGB, the least performed algorithm is LR. But the common problem that we notice is that all three algorithms have a problem when it comes to handling a drastic change. For example, if you see Figure 5.6 wits easy to see the algorithm is not predicting well when drastic change happens on the dataset the same thing happens on all algorithms.

The downside of each algorithm is, it has a problem when it comes to noticing a drastic change in price. If we see the final prediction graphs for all three algorithms (Figure 5.20), they fail to notice a drastic change whether too high or low (for example when the price meets a change more than 1000 Ethiopian birr in just a day) the algorithms fail to recognize.

X. Conclusion and Future Work

When we work on prediction algorithm and prediction android app which is ECPP our aim was to do something new and useful for Ethiopian coffee market because coffee and sesame is a backbone for Ethiopian economy not only that we are the largest exporter in Africa but also the fifth largest exporter in the world. So, our aim in this paper was to be able to provide technological support for this valuable market in Ethiopia.

In our research paper, we successfully did prediction using LSTM, XGB, and LR on two of Ethiopian commodity exchange market items which are Coffee and sesame. We are able to achieve accuracy more than 90% in most of the prediction algorithms which we think good result when it comes to prediction. In order to get this much accuracy, we use some tricks like in all algorithms we implement automating parameter tuning with a wide range of possible result, the diss advantage of this method was it took a lot of time to run the algorithm. In order to check the efficiency of the algorithms, we used three methods which are RMSE, MAPE, and R-squared.

Generally, we compare the algorithms based on their performance and the result that we get LSTM performs better when we compare it with other algorithms. We also observed LSTM performs better prediction with very low MAPE when the data set is big. According to the test that we perform using the *coffee* dataset (1540 instance and 11 attributes) and *sesame* dataset (7205 instances and 11 attributes), the sesame dataset performs better on the LSTM algorithm than a coffee dataset. Consequently, the larger the dataset the better to train the algorithm.

In our second-best algorithm, it is going to be XGB. XGB performs well, the result is actually very close to LSTM. The XGB also performs better when the dataset is large when we utilized the loss functions it gets very low when we give the algorithm a largest dataset. When we implemented LR result it performs less when it compares to LSTM and XGB. But the common thing that it shares with the other two algorithms is it also performs better with a larger dataset to train and test the model.

It is believed that, if we be able to work more on the algorithms and the quality of the dataset and also by selecting proper parameter it is possible to achieve better accuracy. Trading is becoming more advance and it is in the stage of it require technological support apart from smart peoples behind every successful trading.

XI. Future Work

Until the end of this paper, our theory is still at a very early stage, there are still many shortcomings in this article to be improved and improvements will be made in the post-study work. Some of the improvement that we have in our mind is as follow:

1. The first and most basic is to use a very good quality dataset for the prediction algorithms.
2. Performing more feature engineering in order to find a better pattern in the trading sequence and choose better features for the prediction because most of the time researchers use opening, closing, and *adj_clossing* prices in order to perform prediction.
3. Studying the cause for market price variation is one strategy to build a better algorithm to adapt the situation, for our case which is agricultural items mostly affected by weather change and international price variation on the items.
4. Choosing a better algorithm which can perform better for this specific scenario. Here we only use

three different algorithms but there are a lot of different algorithms to try. So, it is better to choose a different algorithm and do more comparative analysis and choose a better performing algorithm.

5. Studying the market behaviour. For this specific case, this agricultural item which is coffee and sesame is seasonal crops so we need to study at which season the price goes high and low
6. Building a better algorithm with better efficiency and tuning the parameter until we get the best result is one option to get a better result.
7. In this research paper, we only focused on the next day prediction but we can also do a long-term prediction. This will help to compare the performance of the algorithm for short- and long-term predictions.

XII. REFERENCES

[1]. L. Nunno, "Stock Market Price Prediction Using Linear and Polynomial Regression Models," p. 6.

[2]. "Ethiopian Coffee." http://www.selamta.net/Ethiopian%20Coffee.htm (accessed Mar. 26, 2020).

[3]. "Coffee production in Ethiopia," Wikipedia. Oct. 27, 2019, Accessed: Dec. 26, 2019. Online]. Available: https://en.wikipedia.org/w/index.php?title=Coffee_production_in_Ethiopia&oldid=923265682.

[4]. K. F. Wiersum, T. W. Gole, F. Gatzweiler, J. Volkmann, E. Bognetteau, and O. Wirtu, "CERTIFICATION OF WILD COFFEE IN ETHIOPIA: EXPERIENCES AND CHALLENGES," For. Trees Livelihoods, vol. 18, no. 1, pp. 9–21, Jan. 2008, doi: 10.1080/14728028.2008.9752614.

[5]. "Top 5 Predictive Analytics Models and Algorithms | Logi Analytics Blog," Logi Analytics, Jul. 09, 2019. https://www.logianalytics.com/predictive-analytics/predictive-algorithms-and-models/ (accessed Mar. 26, 2020).

[6]. K. Pahwa and N. Agarwal, "Stock Market Analysis using Supervised Machine Learning," in 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Feb. 2019, pp. 197–200, doi: 10.1109/COMITCon.2019.8862225.

[7]. "Linear Regression and Prediction." http://jukebox.esc13.net/untdeveloper/RM/Stats_Module_5/Stats_Module_56.html (accessed Dec. 27, 2019).

[8]. R. Kelley, "Machine Learning Explained: Algorithms Are Your Friend." https://blog.dataiku.com/machine-learning-explained-algorithms-are-your-friend (accessed Dec. 26, 2019).

[9]. K. Nishida, "Introduction to Extreme Gradient Boosting in Exploratory," Medium, Mar. 21, 2017. https://blog.exploratory.io/introduction-to-extreme-gradient-boosting-in-exploratory-7bbec554ac7 (accessed Dec. 27, 2019).

[10]. J. Brownlee, "A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning," Machine Learning Mastery, Sep. 08, 2016. https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/ (accessed Mar. 26, 2020).

[11]. "eXtreme Gradient Boosting (XGBoost): Better than random forest or gradient boosting | Welcome to my blog." https://liuyanguu.github.io/post/2018/07/09/extreme-gradient-boosting-xgboost-better-than-random-forest-or-gradient-boosting/ (accessed Mar. 26, 2020).

[12]. "FinTech - Machine Learning and Recommenders," Finextra Research, Jan. 17, 2018. https://www.finextra.com/blogposting/14934/fi

ntech---machine-learning-and-recommenders (accessed Mar. 26, 2020).

[13]. S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[14]. “Understanding LSTM Networks -- colah’s blog.” https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed Dec. 31, 2019).

[15]. Y. Dai and Y. Zhang, “Machine Learning in Stock Price Trend Forecasting,” p. 5.

[16]. S. Kalra and J. S. Prasad, “Efficacy of News Sentiment for Stock Market Prediction,” in 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, Feb. 2019, pp. 491–496, doi: 10.1109/COMITCon.2019.8862265.

[17]. S. Dey, Y. Kumar, S. Saha, and S. Basak, Forecasting to Classification: Predicting the direction of stock market price using Xtreme Gradient Boosting. 2016.

[18]. S. Liu, G. Liao, and Y. Ding, “Stock transaction prediction modeling and analysis based on LSTM,” in 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), May 2018, pp. 2787–2790, doi: 10.1109/ICIEA.2018.8398183.

[19]. H. L. Siew and M. J. Nordin, “Regression techniques for the prediction of stock price trend,” in 2012 International Conference on Statistics in Science, Business and Engineering (ICSSBE), Langkawi, Kedah, Malaysia, Sep. 2012, pp. 1–5, doi: 10.1109/ICSSBE.2012.6396535.

[20]. A. Ioanes and R. Tirnovan, “Energy Price Prediction on the Romanian Market using Long Short-Term Memory Networks,” in 2019 54th International Universities Power Engineering Conference (UPEC), Bucharest, Romania, Sep. 2019, pp. 1–5, doi: 10.1109/UPEC.2019.8893550.

[21]. S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, “Stock price prediction using LSTM, RNN and CNN-sliding window model,” in 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, Sep. 2017, pp. 1643–1647, doi: 10.1109/ICACCI.2017.8126078.

[22]. G. G. Moisen, E. A. Freeman, J. A. Blackard, T. S. Frescino, N. E. Zimmermann, and T. C. Edwards, “Predicting tree species presence and basal area in Utah: A comparison of stochastic gradient boosting, generalized additive models, and tree-based methods,” Ecol. Model., vol. 199, no. 2, pp. 176–187, Nov. 2006, doi: 10.1016/j.ecolmodel.2006.05.021.

[23]. H. Singh, “Understanding Gradient Boosting Machines,” Medium, Nov. 04, 2018. https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab (accessed Jan. 20, 2020).

[24]. Y. Zhang and A. Haghani, “A gradient boosting method to improve travel time prediction,” Transp. Res. Part C Emerg. Technol., vol. 58, pp. 308–324, Sep. 2015, doi: 10.1016/j.trc.2015.02.019.

[25]. P. Carmona, F. Climent, and A. Momparler, “Predicting failure in the U.S. banking sector: An extreme gradient boosting approach,” Int. Rev. Econ. Finance, vol. 61, pp. 304–323, May 2019, doi: 10.1016/j.iref.2018.03.008.

[26]. M. Zięba, S. K. Tomczak, and J. M. Tomczak, “Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction,” Expert Syst. Appl., vol. 58, pp. 93–101, Oct. 2016, doi: 10.1016/j.eswa.2016.04.001.

[27]. Y. Wang, Q. Deng, F. Wen, H. Zhou, F. Liu, and X. Yang, “Combined Use of Support Vector Machine and Extreme Gradient Boosting System for Cost Prediction of Ultra High

Voltage Transmission Projects," in 2019 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia), Chengdu, China, May 2019, pp. 3708–3712, doi: 10.1109/ISGT-Asia.2019.8881151.

[28]. "Forecasting Stock Prices using XGBoost - Towards Data Science." https://towardsdatascience.com/forecasting-stock-prices-using-xgboost-a-detailed-walk-through-7817c1ff536a#c34b (accessed Jan. 16, 2020).

[29]. S. Weisberg, Applied Linear Regression. John Wiley & Sons, 2005.

[30]. "Linear Regression Algorithm | Machine Learning Regression Algorithm," R-ALGO Engineering Big Data, Feb. 02, 2018. https://www.engineeringbigdata.com/linear-regression-algorithm/ (accessed Mar. 27, 2020).

[31]. R. Gandhi, "Introduction to Machine Learning Algorithms: Linear Regression," Medium, May 28, 2018. https://towardsdatascience.com/introduction-to-machine-learning-algorithms-linear-regression-14c4e325882a (accessed Mar. 27, 2020).

Cite this article as :

Sh