Food Price Prediction: A comparative study of statistical and machine learning methods

A thesis submitted in partial fulfillment of the requirements of Varendra University for the degree of BSc Engineering in CSE.

May 2019

Submitted by

ID: 152311028

Department of Computer Science and Engineering

Varendra University

Rajshahi, Bangladesh

Supervised by

Nafi-Us-Sabbir Sabith

Lecturer

Department of Computer Science and Engineering

Varendra University

Rajshahi, Bangladesh

Declaration

I do declare that this thesis "Food Price Prediction: A comparative study of statistical and machine learning methods" is developed by me. This project is submitted to the Department of Computer Science and Engineering, Varendra University, Rajshahi for the partial fulfillment of the degree of B.Sc. in CSE honors is exclusively my original work. This work has not been submitted in any other university or institute for any degree of honors.

Mst. Nishi Khatun

Id: 152311028

Batch: 9th

Semester: 12th

May 2019

Acknowledgement

I take this occasion to thank God, almighty for blessing me with his grace and taking my endeavor to a successful culmination. I extend my sincere and heartfelt thanks to my supervisor Nafi-us-Sabbir for providing me with the right guidance and advice at the crucial junctures and for showing me the right way. I extend my sincere thanks to our respected Head of the department Prof. Dr. Md. Al Mamun, for allowing me to use the facilities available. I would like to thank the other faculty members also, at this occasion. Last but not the least, I would like to thank my friends and family for Support and encouragement they have given me during the course of my work.

Abstract

Agriculture sector is the primary source of food supply of all the countries in the world. Bangladesh is the producer country of major crops such as Rice, Wheat and oil. Data mining is study of extracting useful information from the data. The present study focus on the application of data mining techniques to predict future production of crop such as Rice with respect to various climatic conditions observed in last 12 years (2006-2018) in Bangladesh. The data mining algorithms –Gaussian Process Regression, Support Vector Regression (SMOReg) and Multi-Layer Perceptron are discussed in this paper.

Predicting real values is also an important topic for machine learning. Most of the problems that humans learn in real life, such as sporting abilities, are continuous. Dynamic control is one such problem which is the subject of research in machine learning. In machine learning, most research has been done for classification, where the single the predicted feature is nominal or discrete. Regression differs from classification in that the output or predicted feature in regression problems is continuous.

In this paper we will apply these algorithms for prediction of crop production and find the accuracies of algorithms to compare which one gives better result. This research study will add to the literature in the area of technology development that will handle the fluctuations in the prices and will support the suppliers in a useful manner. This application model will help consumers and suppliers to forecast food price.

Table of Contents

Chapter 1: Introduction ………………………………………….. 09-11

1.1 Motivation 09

1.2 The necessity of solution 10

1.3 Approach 10

1.2 Outline 11

Chapter 2: Literature Review …………………………………….. 12-15

Chapter 3: Problem Formulation ……………………………………… 16-17

3.1 Determining Different Factors that impact food price 16

3.2 Deduce the relationship between food price and factors 16

3.3 Time Series Analysis 16

Chapter 4: Theoretical Framework………………………………………… 18-31

4.1 Data Mining 18

4.2 Machine Learning 20

4.3 Regression 22

4.4 Support Vector Machine 22

4.5 Artificial Neural Network 23

4.6 Algorithm Considered 25

4.6.1 Arima model 25

4.6.2 Linear Regression 26

4.6.3 Gaussian Process Regression 26

4.6.4 Bry Boschan 27

4.6.5 Vector auto Regression (VaR) 28

4.6.6 Stochastic Gradient Descent (SGD) 29

4.6.7 Support Vector Regerssion 30

4.6.8 Multilayer Perceptron 30

Chapter 5: Approach ………………………………….… **32-43**

5.1 Input dataset and pre processing 32

5.1.1 About the dataset 32

5.1.1.1 Timeline of the dataset 32

5.1.1.2 Size of the dataset 32

5.1.1.3 Covered items 32

5.1.1.4 Covered divisions 32

5.1.2 Pre-processing the dataset 33

5.1.2.1 Cleaning the dataset 33

5.1.2.2 Division-wise categorizing 33

5.2 Model Buildup 36

5.2.1 Target selection 36

5.2.2 Basic Parameters 36

5.2.3 Output 38

5.2.4 Advanced Configuration 38

5.3 Model Training 40

5.3.1 Gaussian Regression 41

5.3.2 SMOReg 41

5.3.3 MLP 41

5.4 Model Validation Measures 42

5.4.1 Mean Absolute Deviation (MAD) 42

5.4.2 Mean Square Error (MSE) 42

5.4.3 Root Mean Square Error (RMSE) 42

5.4.4 Mean Absolute Percentage Error (MAPE) 43

Chapter 6: Result ………………………………………………………… 44-45

6.1 Result and accuracy of model 1 44

6.2 Result and accuracy of model 2 44

6.3 Result and accuracy of model 3 45

6.4 Comparison 45

Chapter 7: Conclusion and Future Scope ………………………………… 46

Challenged faced 46

Future Working Scope 47

References ……………………………………………………………… 48-49

Chapter 1

Introduction

1.1 Motivation

The price fluctuation risk of agricultural products has become one of the main risks faced by agricultural producers. The objective of implementing market risk management is to measure and assess accurately the sizes and degrees of risk involving agricultural products.

Agricultural market risk (which also covers price risk), refers to the uncertainties of the expected returns an agricultural producer faces, in relation to certain production factors, to be able to produce the final product during the process of commercialization. These uncertainties exist because of market price fluctuation, making it difficult to give a probability value of possible circumstances. In the market economy, agricultural market price fluctuation is normal, but once the fluctuation margin exceeds the expectations of production operators, risk will exist. Currently, research on agricultural market risk is focused on the level of theoretical analysis and discussion of agricultural market risk characteristics, as well as the formulation of preventive measures.

Bangladesh being a developing country, always tries to control the market prices of coarse rice for stable political and social condition. The main aim of this paper is to find out appropriate deterministic time series model using the latest selection criteria that could best describe the coarse rice price pattern in Bangladesh during the time period May 2006 to September 2018. A total of 438 secondary data on monthly wholesale prices of coarse rice from May 2006 to September 2018 have been used for time series analysis.

Bangladesh is predominantly an agriculture based country with more than 149 million people living on 14.84 million hectares of land. Economic development of the country is based mainly on agriculture as the contribution of agriculture sector in GDP is 16.03% [1]. Rice is one of the leading food crops to fulfill the demand of carbohydrate in Bangladesh. At present rice is cultivated in 10.58 million hectares of land. It alone constitutes the lion share (96%) of total food grain produced in Bangladesh (BBS, 2010). In Bangladesh 63% of the labor force is directly engaged in agriculture and 78% of total cropland diverted to rice production (BER, 2010). Since independence food autarky of Bangladesh has become dependent on rice production. All the five year plans had especial emphasis on the production of rice.

Forecasting price of the important commodities is very essential for the policy makers of a country. Bangladesh as a developing country where approximately 40% of the people are living below the poverty line can take necessary action if it can form an idea about the future price of the selected commodities. For many countries, especially developing countries, primary commodities remain an important source of export earnings, and commodity price movements have a major impact on overall macroeconomic performance.

Evaluation of the price pattern of rice in Bangladesh may serve as an aid for policy makers in taking decision regarding production, procurement, export, import etc. To reveal the price pattern and to make the best forecasts of coarse rice price in Bangladesh appropriate time series models that best describe the observed data successfully are necessary.

1.2 The necessity of solution

predicting food price will help farmers, consumers, and business entities [2]. Farmers can get an insight of the price of different commodities in future and look to grow the profitable commodities. They can also avoid the bitter experience of not getting the desired market price. The consumers can foresee the market risks and plan accordingly. For business entities, the solution provides a new dimension in planning their activities.

1.3 Approach

A public dataset from the World Food Program (WFP) was used. The dataset contained prices of various commodities from around the world and was updated monthly. The monthly fluctuation of food prices was evident and this presented a new dimension in conducting this research work.

The dataset needed multi staged preprocessing in order to be applied as input to algorithms. Firstly, unnecessary and redundant attributes were discarded. Then it was grouped primarily on the basis of division and then on the basis of commodity type. Then a split of training data and test data was done to validate the models we propose here. Then the models were trained and the result was compared with actual prices of test data.

1.4 Outline

The book has been organized and categorized over 5 chapters. Each chapter is a continuation of its predecessor and a smooth flow of information has been maintained throughout the chapters.

Chapter 2 contains the background study for this research work.

Chapter 3 addresses the problem and its formulation in terms of computer science terms and jargons.

A theoretical framework has been molded in the following chapter, chapter 4.

A novel system model is proposed in the next chapter, chapter 5. The chapter enlists information about /metadata of the dataset. The four different models, each with its unique blend of characteristics had been tried and tested upon. The theoretical aspect of these models and their implementation in regards of this research work in documented in this chapter.

Chapter 6 is all about the results of this research work.

Foreclosing remarks and challenges, as well as future working score has been discussed briefly in the final chapter.

Chapter 2

Literature Review

Research papers focused on agricultural price forecasting, consumer food prices forecasting, real estate price prediction, electricity price forecasting, foreign currency exchange rate forecasting and stock market prediction were studied with a view to gaining in-depth domain knowledge and make a reasoning of what had to be done.

T. Gubanova & et al forecasted organic food prices using three forecasting methods from three different family of algorithms to deal with data featuring seasonal variation [3]. A seasonal autoregressive (AR) model was chosen out of the ARMA class, additive version of the Holt-Winters (HW) exponential smoothing was chosen out of the exponential smoothing family and a spectral decomposition (SD) were implemented to operate in a fully automatic way in order to provide the level ground for their competition. They found AR to be broadly the best forecasting method as compared to both SD and HW methods, for all produce items and all horizons. HW appears to be the best forecasting method for the ten-days-ahead forecast horizon. SD outperforms HW for medium and relatively long-term forecasts. The best performing method was found among these three industry-oriented forecasting techniques. Based on both the quantile analysis and the Giacomini-White test, seasonal auto regression is the best forecasting method, compared to spectral decomposition and the Holt-Winters exponential smoothing for all produce and all horizons. Intriguingly, they identified directions for future research as follows.

- Adaptation of forecasting methods for cases when the data are unevenly-spaced.

- Missing data are a common problem not only for agricultural data but for economic data in general. More effective techniques need to be implemented instead of linear spline interpolation used in the present research.

- Instead of applying a forecasting method to one commodity at a time, prices for a group of products can be forecast jointly, in order to account for an effect of substitution amongst commodities. Spectral decomposition and multivariate ARMA allow to conduct such a kind of analysis.

- Combining several methods. Even though the seasonal ARMA was found to be the best performing method, ARMA forecasts can be combined with those from the Holt-Winters method and spectral decomposition to further improve the forecast quality.

B.Guha & et al [4] applied ARIMA model to forecast price of Gold with the basic assumption that it follows a perfectly linear pattern. The value of Durbin-Watson (DW) was 0.091 for the sample data of the Gold price from November 2003 to January 2014 which indicates that the data is suitable for time-series analysis. But they pointed out that There are certain limitations in forecasting a data with ARIMA modeling. This technique is used for short run only, to detect small variation in the data. In case of sudden change, in the data set (when the variation is large) in case of change in government policies or economic instability (structural break) etc. it becomes difficult to capture the exact change, hence this model becomes ineffective to forecast in this scenario more over the forecasting with this method is based on assumption of linear historic data but there is no evidence that the gold price is linear in nature. An important remark stated that implementing non-linear forecasting techniques using soft computing techniques can be considered with less white noise term.

J.Wang & et al did work on Cycle Phase Identification and Factors Influencing the Agricultural

Commodity Price Cycle in China from the Evidence of Cereal Prices [5]. They applied seasonal adjustments to monthly cereal price data then identified and replaced extreme outliers. They made use of Bry-Boschan algorithm of survival analysis model. The result shows not only the price cycles of six kinds of cereal specified by using the Bry-Boschan algorithm but also the duration of the crude oil price cycle, which is later used as the covariate variable in the survival analysis to explain the fluctuation in the cereal price cycle. Their paper only studied the effect of crude oil prices on the hazard rate for China’s cereal price cycles. However, they remarked that other co-variables, such as the auction time for the cereal market, transaction volume, weather conditions, and so on, are also likely to produce effects on the hazard rate for China’s cereal price cycles.

G.Li & et al performed Short-Term Price Forecasting For Agro-products Using Artificial Neural Networks(ANN). [6] They applied ARIMA and ANN model and showed the comparison between this two model. The time interval of data was shorter, so the accuracy was higher. Results showed the accuracy of daily price forecasting is the best. Next higher accuracy is weekly price forecasting. Their results exhibited the fact that The ANN model is more suitable for forecasting ahead of one cycle. Time series model and ANN model are equally effective in accuracy for forecasting ahead of one day, one week and one month, but ANN model is

better than time series model. The accuracy of ANN model for three types of forecasting is more than 80%, and daily price forecasting is even more than 90%.

T.R.Cook & et al experimented in Macroeconomic Indicator Forecasting with Deep Neural Networks.[7] Fully Connected Neural Networks, Convolutional Neural Networks, Long Short

Term Memory (LSTM) Networks and Encoder Decoder Networks were used to train and test on 0-4 quarter prediction horizons.

L.Nuno used Linear Regression, Stochastic Gradient Descent (SGD) and Support Vector Regression (SVR) with polynomial and RBF kernel to predict stock market prices [8]. The results for 180-day price predictions were chosen to provide insight onto the performance of these algorithms for a longer period of time. Overall, SVR with the RBF kernel performed the best, but it is interesting to note that SVR with the polynomial performed better in comparison with the rest of the algorithms on these longer time frames. Linear regression performed very poorly when its window size was small for long-term price prediction.

R.Kumar & et al worked on time series forecasting of Nifty stock data by applying the four different algorithms to same dataset-

i. Gaussian processes,
ii. linear regression,
iii. Multilayer Perceptron and
iv. SMOreg[9].

They examined and applied different forecasting techniques by using the Weka tool and compared various prediction functions, and found that SMO regression function offer the ability to predict the stock price of NSE more accurately than the other functions such as Gaussian processes, linear regression, multilayer perceptron. This analysis can be used to reduce the error percentage in predicting the future stock prices. It increases the chances for the investors to predict the prices more accurately by reduced error percentage and hence increased profit in share markets. More Accurate results can be found if we will take data of more duration. It will help to check deviations in Values and predict more accurate results.

J.Harris conducted a survey on machine learning approach to Forecasting consumer food prices [10]. Experimented with four algorithms from four different classes –

i. Multivariate Regression: Linear Regression
ii. Artificial Neural Networks: Multilayer Perceptron
iii. Support Vector Machine: SMOreg
iv. Decision Tree: M5P Tree

They Determined the top performing model of the three models assessed (Holt-Winters, Food Price Report, and Financial Futures-based Markets). Overall the predictions were extremely close with 5 of the 8 categories having achieved error rates under 2% and only 1 category with a lowest error rate above 3%. The Financial Future-Market based model managed an average error rate of 2.8%, 1.5% below the Food Price Report model and 10% below the benchmark model, to solidify its position as the top 40 performing model. The data from the Financial Futures-Market model, as the top performing model were then used to evaluate the top performing the technique and the results are resented in the following section.

The evaluation of the top performing techniques does leave some questions unanswered such as what caused the Holt-Winters model to outperform the Linear Regression technique. The Holt-Winters model posted an average error of 8 while the Linear Regression technique was only able to achieve 9.8 in the top performing model. Additionally, the Average error rate of the M5P Tree technique is incredibly close to that of the top performing Multilayer Perceptron even though the M5P tree technique was only able to secure a top performance in a single category, vegetables.

An application of SVM regression was discussed in (Fernandez, 1999). The problem was time series prediction. The approach taken was the use of SVM regression to model the dynamics of the time series and subsequently predict future values of the series using the constructed model. Instead of using the standard SVM regression formulation described above, a variation developed in (Scholkopf et al., 1998) was used. Using this variation the ε parameter of the SVM regression loss function (see above) is automatically estimated. Furthermore, (Fernandez, 1999) used an approach to learning which is different from the standard one: instead of developing one global regression model from all the available training data, (Fernandez, 1999) develops a number of SVM regression models, each one trained using only part of the initial training data. The idea, which has been suggested in (Bottou and Vapnik, 1992), is to split the initial training data set into parts, each part consisting only of training data that are close to each other (in a Euclidean distance sense). Then a "local" SVM is trained for each subset of the data. The claim in (Bottou and Vapnik, 1992) is that such an approach can lead to a number of simple (low complexity, in the SLT sense outlined above) learning machines, instead of a single machine that is required to fit all data. In (Fernandez, 1999) each of the individual SVM machines had its ε parameter estimated independently. The ε parameter of the SVM loss function is known to be related to the noise of the data (Pontil et al., 1998). So, in a sense, the approach of (Fernandez, 1999) leads to local SVMs each having an ε parameter that depends on the noise of the data in particular regions of the space (instead of a single ε that needs to "model" the noise of all the data). The experiments described in (Fernandez, 1999) show that training many local SVMs instead of one global learning machine leads to significant improvements in performance. In fact, this was also the finding of (Bottou and Vapnik, 1992) who first showed experiments with local learning machines.

Chapter 3

Problem Formulation

3.1 Determining Different factors that impact food price

Food price is a complex and non-linear function of its variables/factors. It is composed of different complex variables as parameters such as – place, time, natural disaster, man-made artificial crisis etc.

3.2 Deduce the relationship between food price and factors

By hand there was no feasible way to come up with a formula that could map the food price as a function to its factors.

So, various machine learning algorithms were used to deduce the complex and non-linear relationship.

3.3 Time Series Analysis

"Time series data" is a special type of data that is made up of a sequence of data points taken at successive equally spaced points in time.

Time series analysis is the process of using statistical techniques to model and explain a time-dependent series of data points. Time series forecasting is the process of using a model to generate predictions (forecasts) for future events based on known past events. Time series data has a natural temporal ordering - this differs from typical data mining/machine learning applications where each data point is an independent example of the concept to be learned, and the ordering of data points within a data set does not matter. Examples of time series applications include: capacity planning, inventory replenishment, sales forecasting and future staffing levels.

To foresee the future and avoid any market risk, prediction and forecasting is essential to *stay ahead of time*.

Chapter 4

Theoretical Framework

The concepts of regression, data mining and different aspects of machine learning is provided here.

4.1 Data mining

Data Mining is defined as extracting information from huge sets of data. In other words, we can say that data mining is the procedure of mining knowledge from data. It is the process of discovering interesting knowledge from large amounts of data stored either in databases, data warehouses, or other information repositories.

“Data Mining”, often also referred to as “Knowledge Discovery in Databases” (KDD), is a young sub-discipline of computer science aiming at the automatic interpretation of large datasets.

The classic definition of knowledge discovery by Fayyad et al. from 1996 describes KDD as “the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data” (Fayyad et al. 1996). Additionally, they define data mining as “a step in the KDD process consisting of applying data analysis and discovery algorithms [11]”. Over the last decade, a wealth of research articles on new data mining techniques has been published, and the field keeps on growing, both in industry and in academia.

Here is the list of steps involved in the KDD process in data mining −

1. **Data Cleaning** − Basically in this step, the noise and inconsistent data are removed.

2. **Data Integration** − Generally, in this step, multiple data sources are combined.

3. **Data Selection** − Basically, in this step, data relevant to the analysis task are retrieved from the database.

4. **Data Transformation** −In this step, data is transformed into forms appropriate for mining. Also, by performing summary or aggregation operations.

5. **Data Mining** − Generally, in this, intelligent methods are applied in order to extract data patterns.

6. **Pattern Evaluation** − Basically in this step, data patterns are evaluated.

7. **Knowledge Presentation** − Generally, in this step, knowledge is represented.

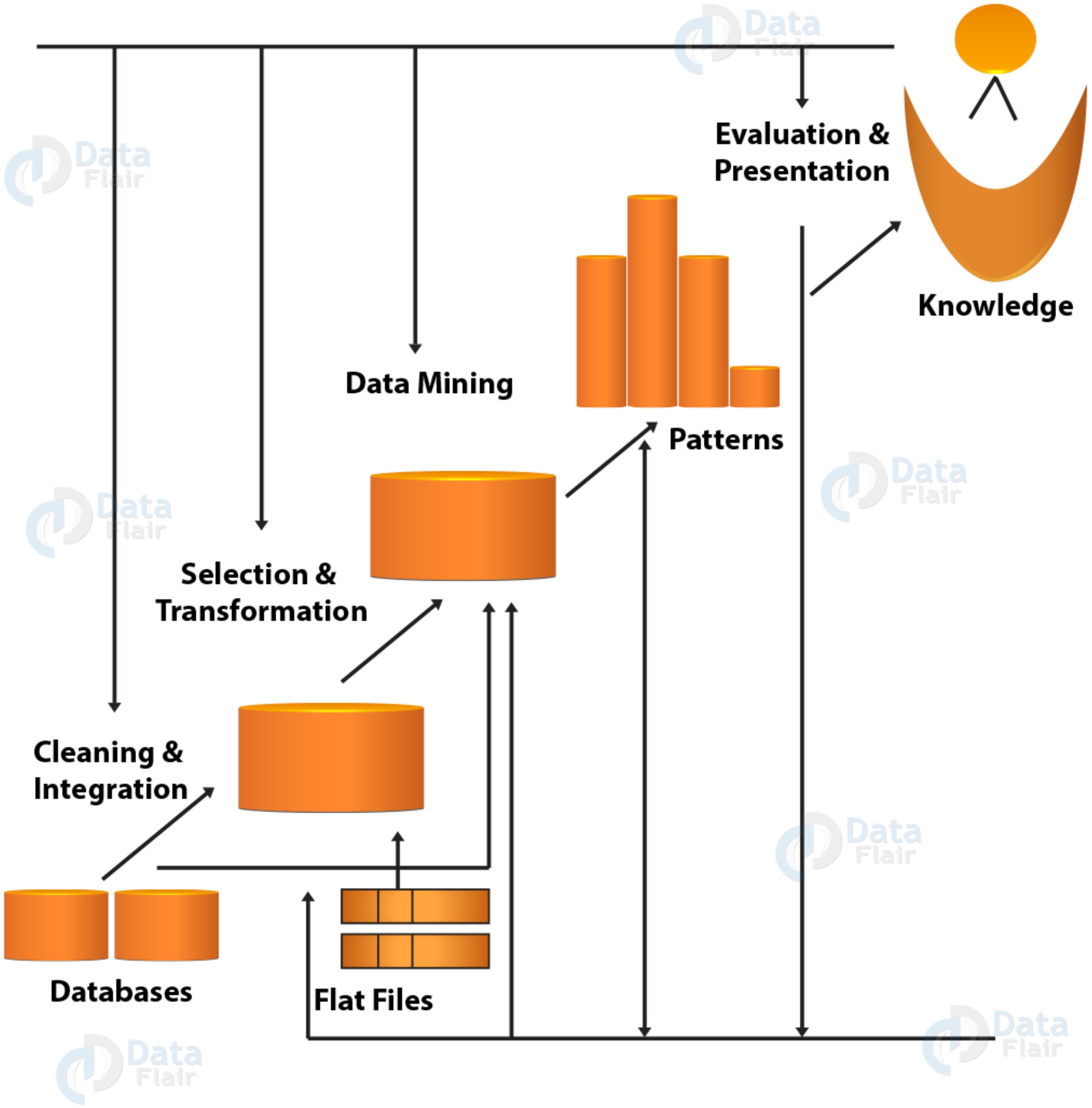


Figure 4.1: Aspects of Data Mining and Knowledge Discovery

Five Major Elements of Data Mining

i. Extract, transform and load transaction data onto the data warehouse system.
ii. Basically, it stores and manages the data in a multidimensional database system.
iii. Generally, provide data access to business analysts and information technology professionals.

iv. As basically it analyzes the data by application software.
v. Basically, it shows the data in a useful format, such as a graph or table.

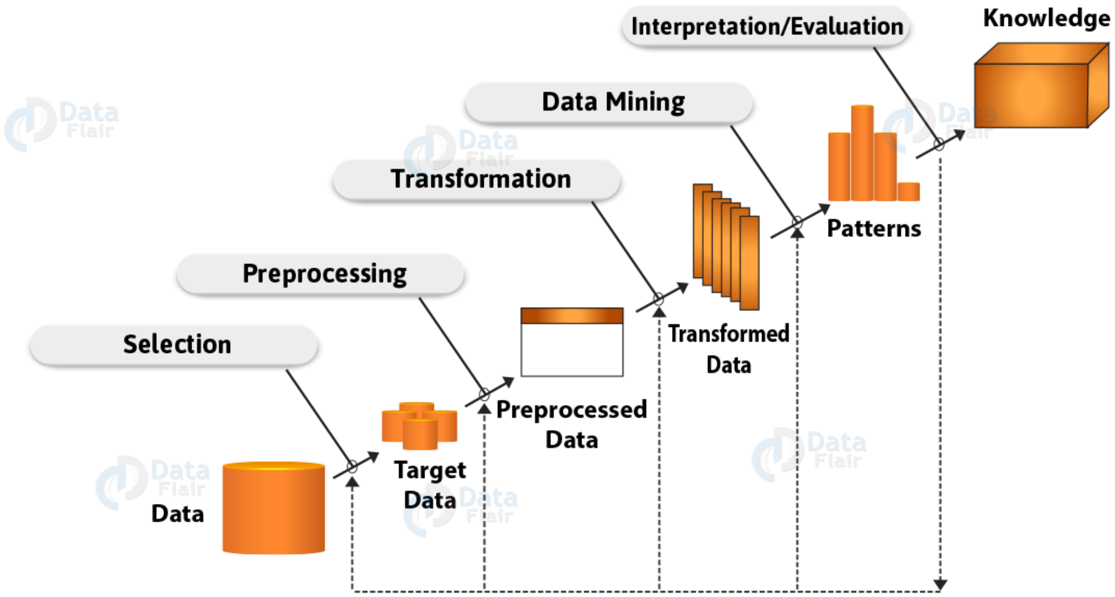


Figure 4.2: Elements of Data Mining and Knowledge Discovery

4.2 Machine Learning

Machine learning is a branch of artificial intelligence that is concerned with building systems that require minimal human intervention in order to learn data and make accurate predictions [11]. According to Breiman [12] and Hall et al [11], in contrast to many statistical approaches, which can value inference over prediction, machine learning focuses on prediction accuracy.

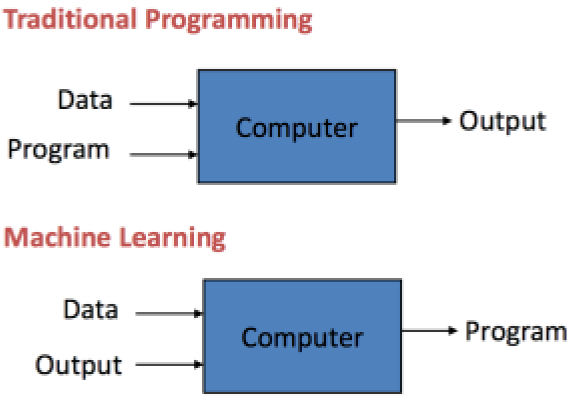


Figure 4.3: Traditional vs Machine Learning

Machine learning helps eliminate the static, fixed and strict approach of well-structured programming which usually provides for either poor optimization or non-efficient use of memory space and time-based factors [12]. Machine learning is composed of two phases, namely, a learning phase and a prediction phase as shown in Fig. 1. The learning phase involves the following: 1) preprocessing (normalization, reduction, data cleansing); 2) learning (supervised, unsupervised and reinforcement); 3) error analysis (precision/recall, over fitting, test/cross validation etc.); and 4) model building [13]. The prediction phase takes the output of the learning phase, which is the model to predict new data sets.

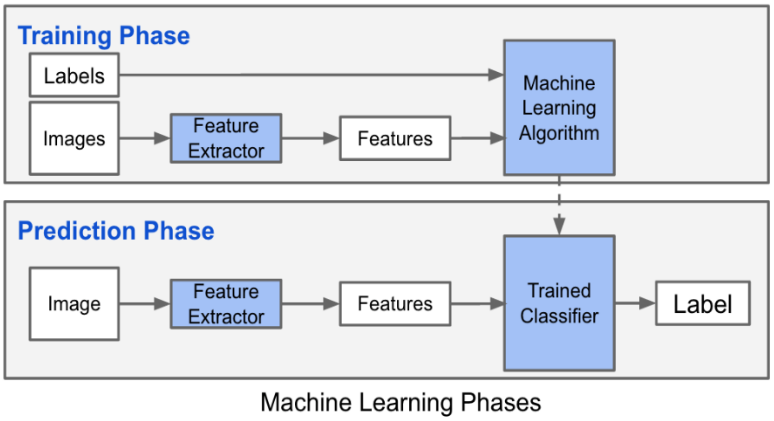
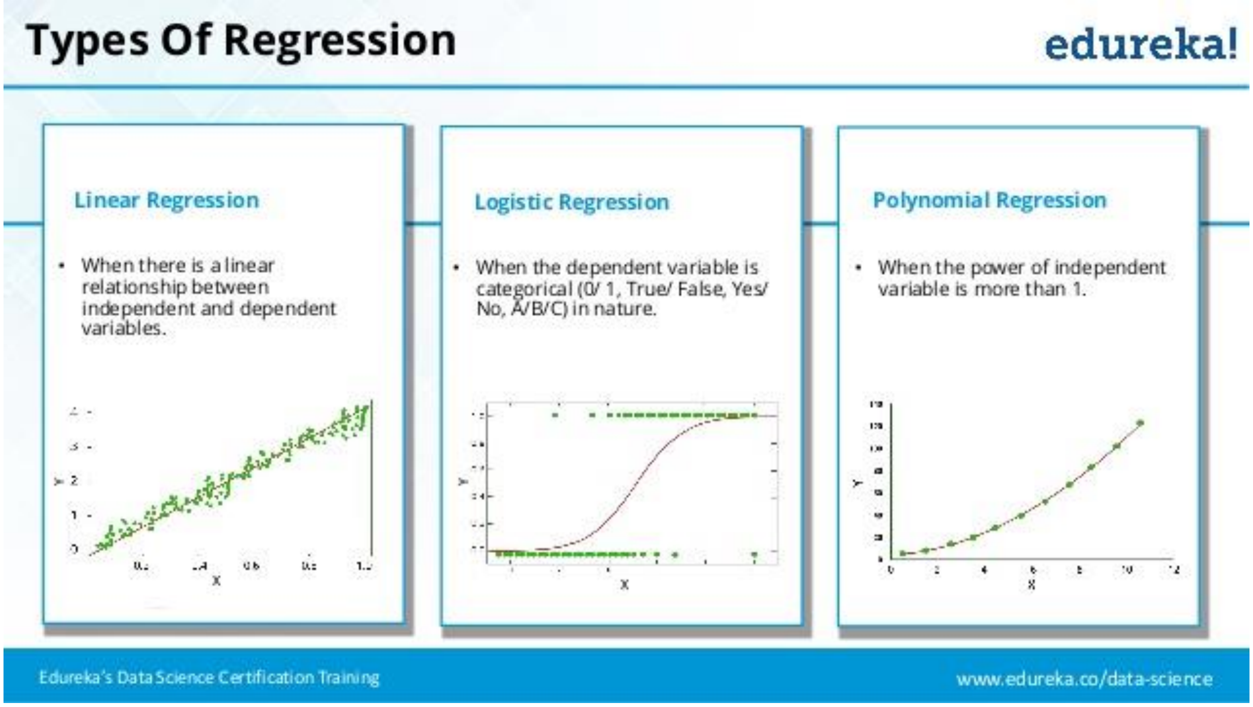


Figure 4.4: Machine Learning Phases

The predicted data helps management or decision makers make informed decisions that are further used to build a knowledge discovery database [13].

4.3 Regression

Predicting or learning numeric features is called regression in the statistical literature, and it is the subject of research in both machine learning and statistics [14]. Regression is important for many applications, since lots of real life problems can be modeled as regression problems.



Types Of Regression

edureka!

Linear Regression

- When there is a linear relationship between independent and dependent variables.

Logistic Regression

- When the dependent variable is categorical (0/ 1, True/ False, Yes/ No, A/B/C) in nature.

Polynomial Regression

- When the power of independent variable is more than 1.

Edureka's Data Science Certification Training

www.edureka.co/data-science

Figure 4.5: Regression Types

4.4 Support Vector Machine

Support Vector Machines (SVM) have been recently developed in the framework of statistical learning theory (Vapnik, 1998) (Cortes and Vapnik, 1995), and have been successfully applied to a number of applications, ranging from time series prediction (Fernandez, 1999), to face recognition (Tefas et al., 1999), to biological data processing for medical diagnosis (Veropoulos et al., 1999) [15]. Their theoretical foundations and their experimental success encourage further research on their characteristics, as well as their further use. [16]

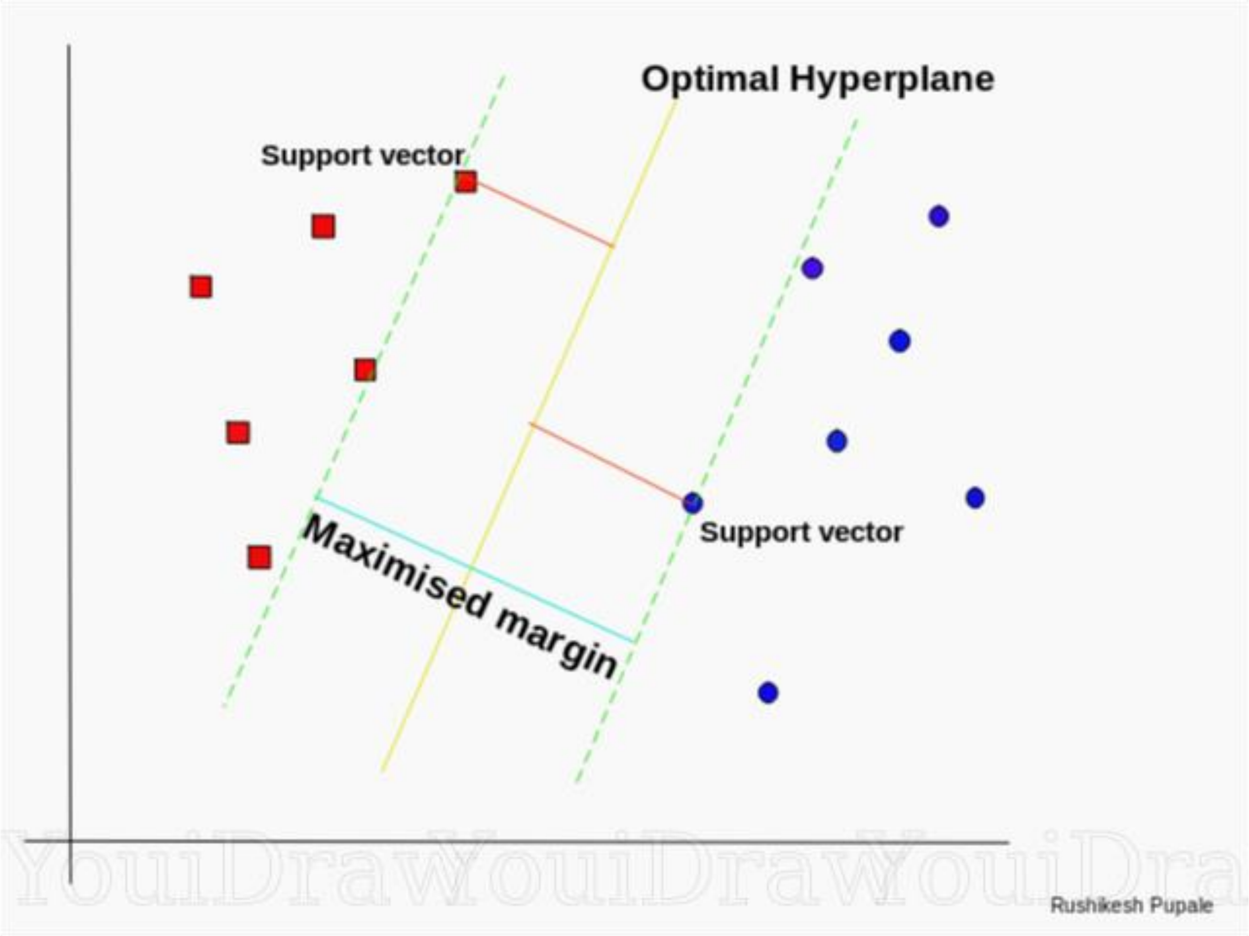


Figure 4.6: Optimal Hyperplane using the SVM algorithm

SVM tries to make a decision boundary in such a way that the separation between the two classes is as wide as possible

4.5 Artificial Neural Networks

Artificial neural networks (ANNs) have emerged as a powerful statistical modeling technique. ANNs provide an attractive alternative tool for both researchers and practitioners. They can detect the underlying functional relationships within a set of data and perform tasks such as pattern recognition, classification, evaluation, modeling, prediction and control. Neural networks are particularly well suited to finding accurate solutions in an environment characterized by complex, noisy, irrelevant or partial information [17]. In the finance and economics fields, many price behaviors have memory and may be modeled more accurately with techniques other than the traditional linear statistical methods. Some emerging nonlinear methods, in particular neural networks, are being increasingly applied in finance and economics since neural networks have more general functional forms than those that can be effectively dealt with by well-developed statistical methods. The application involves the interaction of many diverse variables that are highly correlated, frequently assumed to be nonlinear, unclearly related, and too complex

to be described by a mathematical model.

Several distinguishing features of ANNs make them valuable and attractive in forecasting.

First, ANNs are nonlinear data-driven. They are capable of performing nonlinear modeling without a priori knowledge about the relationships between input and output variables. Thus they are more general and flexible modeling tools for forecasting. The non-parametric ANN model may be preferred over traditional parametric statistical models in situations where the input data do not meet the assumptions required by the parametric model, or when large outliers are evident in the dataset.

Second, ANNs are universal functional approximators. It has been shown that a network can approximate any continuous function to any desired accuracy.48,49 ANNs have more general and flexible functional forms than the traditional statistical methods can effectively deal with.

Third, ANNs can generalize. After learning the data presented to them, ANNs can often correctly infer the unseen part of a population even if the sample data contain noisy information. As forecasting is performed via prediction of future behavior (the unseen part) from examples of past behavior, it is an ideal application area for neural networks, at least in principle.

These unique features make ANNs valuable for solving many practical forecasting problems.

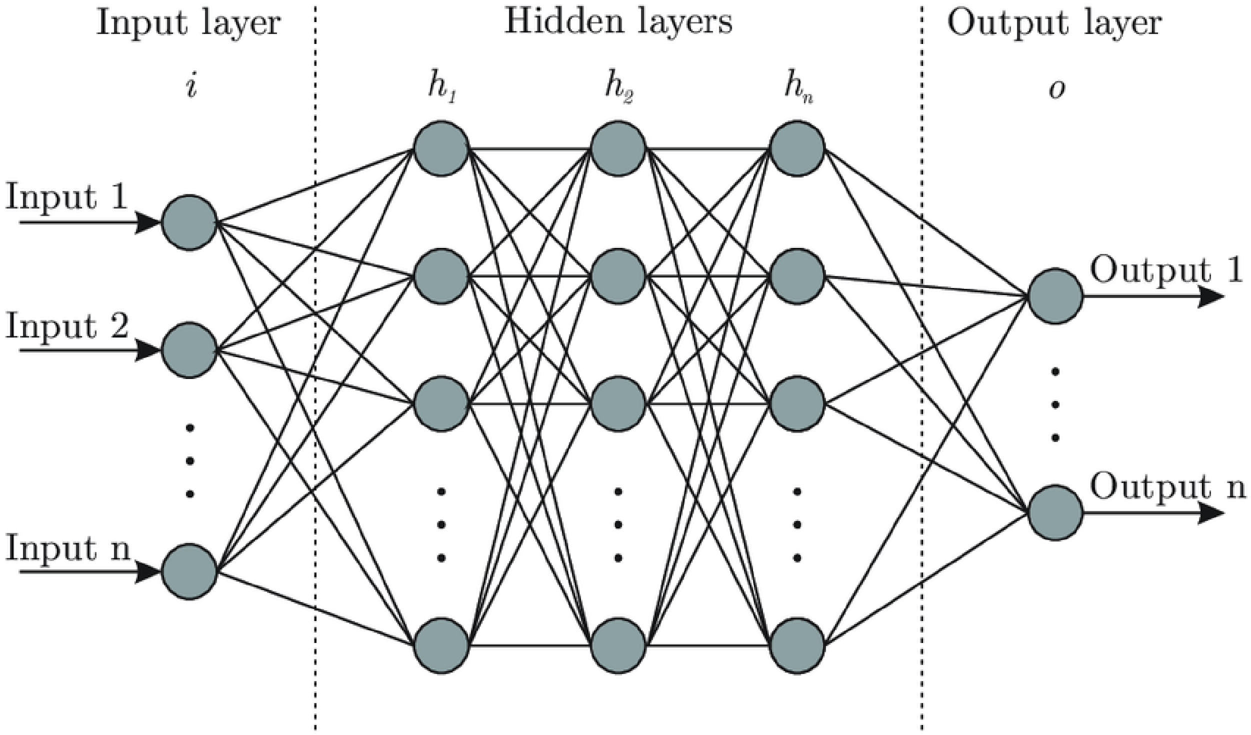


Figure 4.7: ANN architecture

Any time series forecasting model assumes that there is an underlying process from which data are generated and the future value of a time series is solely determined by the past and current observations. Neural networks are able to capture the underlying pattern or autocorrelation structure within a time series even when the underlying law governing the system is unknown or too complex to describe.

4.6 Algorithms considered

4.6.1. ARIMA model

ARIMA stands for Auto Regressive Integrated Moving Average. It's a way of modelling time series data for forecasting (i.e., for predicting future points in the series), in such a way that:

- a pattern of growth/decline in the data is accounted for (hence the "auto-regressive" part)
- the rate of change of the growth/decline in the data is accounted for (hence the "integrated" part)
- noise between consecutive time points is accounted for (hence the "moving average" part)

ARIMA is usually superior to exponential smoothing techniques when the data is reasonably long and the correlation between past observations is stable. If the data is short or highly volatile, then some smoothing method may perform better. If you do not have at least 38 data points, you should consider some other method than ARIMA.

An ARIMA model can be understood by outlining each of its components as follows:

- *Autoregression (AR)* refers to a model that shows a changing variable that regresses on its own lagged, or prior, values.
- *Integrated (I)* represents the differencing of raw observations to allow for the time series to become stationary, i.e., data values are replaced by the difference between the data values and the previous values.
- *Moving average (MA)* incorporates the dependency between an observation and a residual error from a moving average model applied to lagged observations.

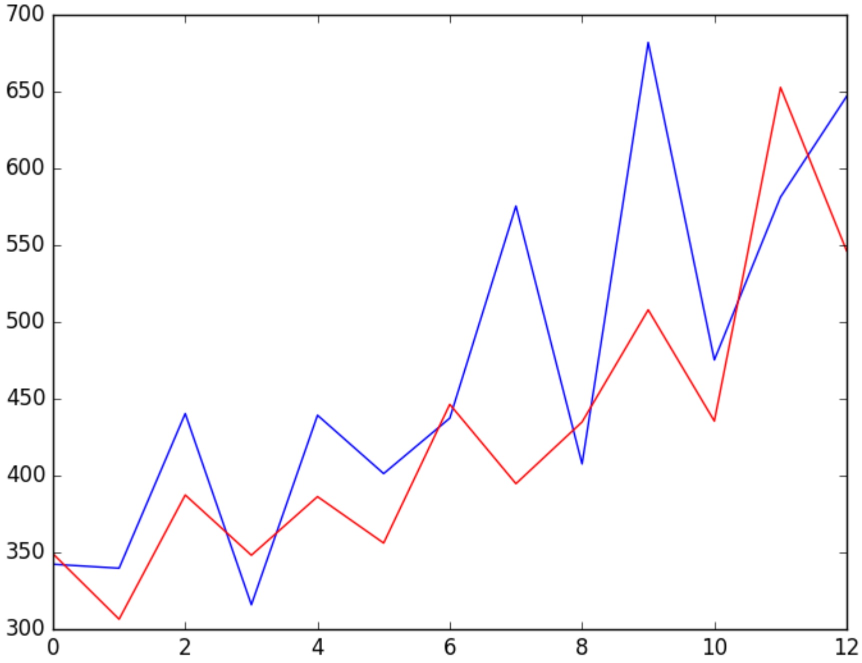


Figure 4.8: ARIMA model

4.6.2 Linear regression

Linear regression was less sensitive to normalization techniques as opposed to the polynomial regression techniques. Some plausible results were appearing early on in the study even when a small number of features were used without normalization, while this caused the polynomial regression models to overflow. Linear regression also provided plausible results after normalization with no parameter tuning required due to its simplified model, although the accuracy was less than would be desired if relying on the results for portfolio building.

The **linear regression model** describes the dependent variable with a straight line that is defined by the **equation** Y = a + b × X, where a is the y-intersect of the line, **and** b is its slope. ... The proper interpretation of the **regression** coefficient thus requires attention to the units of measurement.

A linear regression refers to a regression model that is completely made up of linear variables. Beginning with the simple case, Single Variable Linear Regression is a technique used to model the relationship between a single input independent variable (feature variable) and an output dependent variable using a linear model i.e a line.

A few key points about Linear Regression:

- Fast and easy to model and is particularly useful when the relationship to be modeled is not extremely complex and if you don't have a lot of data.
- Very intuitive to understand and interpret.
- Linear Regression is very sensitive to outliers.

4.6.3 Gaussian process regression

A machine-learning algorithm that involves a Gaussian process uses lazy learning and a measure of the similarity between points (the kernel function) to predict the value for an unseen point from training data.

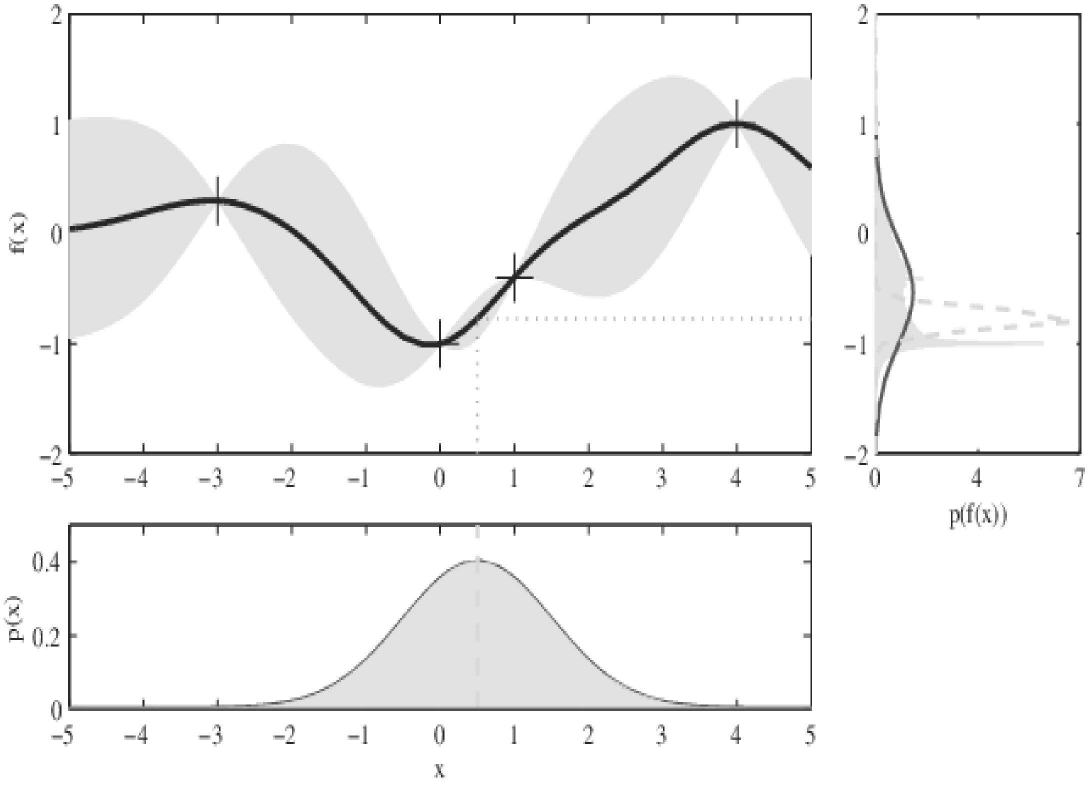


Figure 4.9: Gaussian process regression

4.6.4 Bry-Boschan

The Bry-Boschan (BB) procedure (Bry and Boschan, 1971) simplifies this methodology, providing an algorithm to determine turning points in a single monthly series, such as real GDP. (cyclical) component. In business cycle research the most commonly applied detrending technique is the Hodrick Prescott (1997) filter.

The Bry-Boschan algorithm is a complex multi-step process which (for monthly data)

i. Replaces "outliers" in the data (those too far from a preliminary trend-cycle).
ii. Locates preliminary turning points by finding local maxima and minima of the adjusted trend-cycle.
iii. Eliminates consecutive "peaks" and consecutive "troughs", by keeping the most extreme in a sequence.
iv. Enforces a minimum cycle length (eliminating the less pronounced peak-trough combination to make that happen).
v. Does some further refinement to make sure "phases" aren't too short.

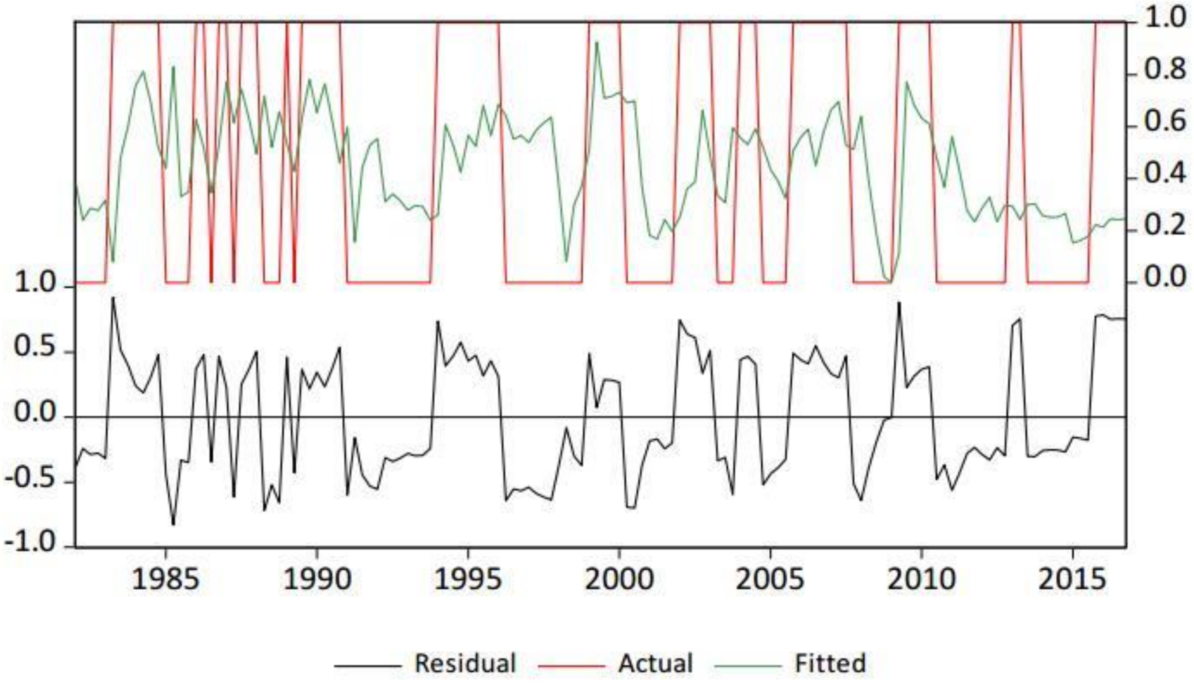


Figure 4.10: Bry-Boschan algorithm

4.6.5 Vector auto Regression (VaR)

In general, quantitative analysis of socio-economic phenomena with a mathematical model must follow certain assumptions. The VaR model includes the following assumptions: a) the market is efficient; b) market volatility is random. From the object of study in this paper, we can see that the agricultural market of China has been proven, in previous studies, to be the weak form efficiency market. Furthermore, China has relatively high degree of openness in fruit market, the price is determined mainly by market supply and demand, and market price fluctuation is random. Therefore, this fruit market can completely satisfy the required assumptions of the VaR model, guaranteeing the possibility that the fruit market risk of China could be calculated using the VaR method.

The vector autoregression (VAR) model is one of the most successful, flexible, and easy to use models for the analysis of multivariate time series. It is a natural extension of the univariate autoregressive model to dynamic multivariate time series. The VAR model has proven to be especially useful for describing the dynamic behavior of economic and financial time series and for forecasting. It often provides superior forecasts to those from univariate time series models and elaborate theory-based simultaneous equations models. Forecasts from VAR models are quite flexible because they can be made conditional on the potential future paths of specified variables in the model. In addition to data description and forecasting, the VAR model is also used for structural inference and policy analysis. In structural analysis, certain assumptions about the causal structure of the data under investigation are imposed, and the resulting causal impacts of unexpected shocks or innovations to specified variables on the variables in the model are summarized. These causal impacts are usually summarized with impulse response functions and forecast error variance decompositions.

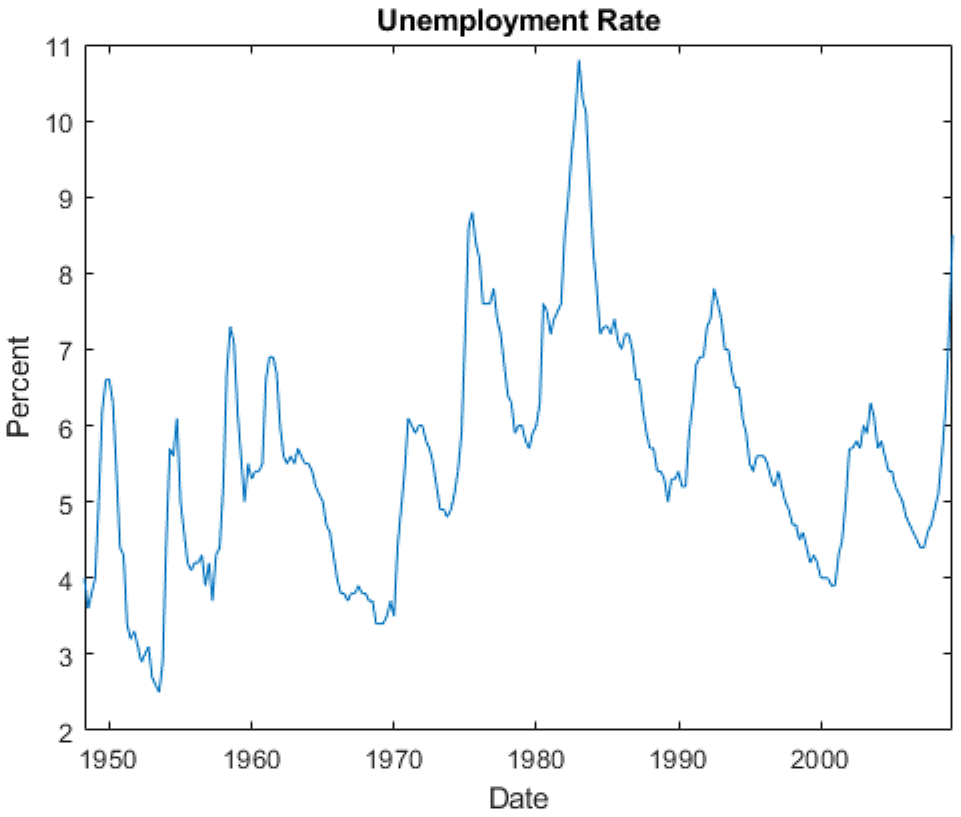


Figure 4.11: Vector auto Regression algorithm

4.6.6 Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent. Stochastic Gradient Descent (**SGD**) is a simple yet very efficient approach to discriminative learning of linear classifiers under convex loss functions such as (linear) Support Vector Machines and Logistic Regression.The advantages of Stochastic Gradient Descent is its Efficiency.

At first, it appeared that Stochastic Gradient Descent would be an appropriate fit to a problem of this type for long term price prediction. However, since the dataset that was used only covered the time period of 2005-2013 the training data could only provide a maximum of $(365 * 8) = 2920$ training samples to be used. Obviously, the stock exchange is not open every day of the year, therefore this number would be significantly lower. This appears to be a problem according to the algorithm's documentation source, since it is only recommended to be used for problems with a training set size of greater than 10,000. SGDRegressor is well suited for regression problems with a large number of training samples ($> 10, 000$), for other problems we recommend Ridge, Lasso, or ElasticNet. Further along in the paper, we will investigate some of the alternatives mentioned above, but this is also an opportunity for future research on linear methods applied to this domain.

4.6.7 Support Vector Regression (SVR) using SMOreg

Sequential minimal optimization (SMO) is an algorithm for solving the quadratic programming (QP) problem that arises during the training of support vector machines (SVM). SMOreg implements the support vector machine for regression. The parameters can be learned using various algorithms. The algorithm is selected by setting the RegOptimizer. The most popular algorithm (RegSMOImproved) is due to Shevade, Keerthi et al and this is the default RegOptimizer. [18]

4.6.8 Multi Layer Perceptron (MLP)

The Multi Layer Perceptron (MLP) algorithm is a powerful form of an Artificial Neural Network that is commonly used for regression. It is a supervised learning algorithm that can be used for both classification and regression for any type of N-dimensional signal.

The MLP algorithm is a very good algorithm to use for the regression and mapping. It can be used to map an N-dimensional input signal to an M-dimensional output signal, this mapping can also be non-linear.

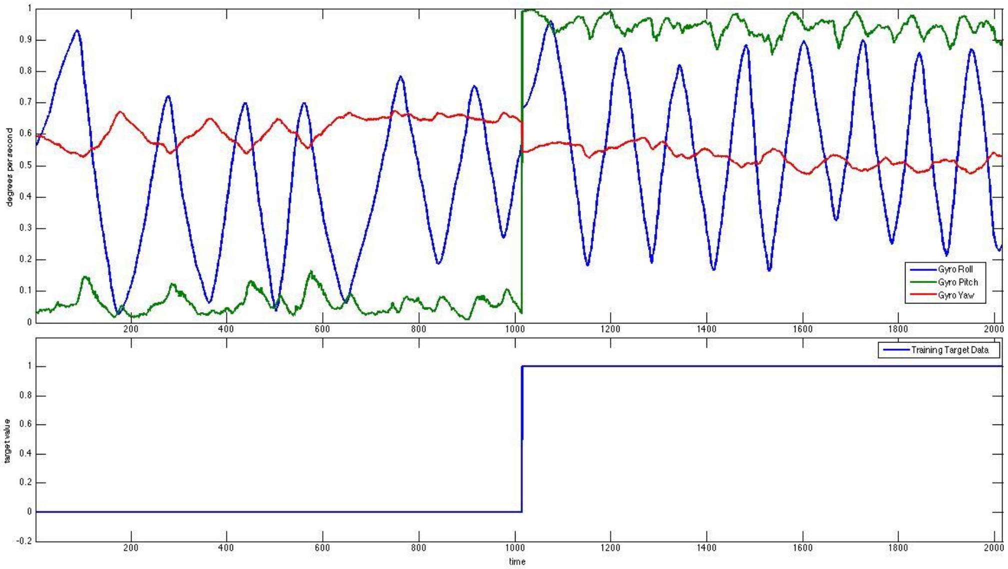


Figure 4.12: Multi Layer Perceptron Regression

The main limitation of the MLP algorithm is that, because of the way it is trained, it cannot guarantee that the minima it stops at during training is the global minima. The MLP algorithm can, therefore, get stuck in a local minimum. One option for (somewhat) mitigating this is to train the MLP algorithm several times, using a different random starting position each time, and then pick the model that results in the best RMS error. The number of random training iterations can be set usingthe setNumRandomTrainingIterations(UINTnumRandomTrainingIterations) method, setting this value to a higher number of random training iterations (i.e. 20) may result in a better classification or regression model, however, this will increase the total training time. Another limitation of the MLP algorithm is that the number of Hidden Neurons must be set by the user, setting this value too low may result in the MLP model under fittingb while setting this value too high may result in the MLP model overfitting.

Chapter 5

Approach

5.1 Input dataset and pre processing

The input dataset and its attributes are discussed in brief. The several pre-processing steps, required to work on just the essential attributes of the dataset are also stated.

5.1.1 About the dataset

Bangladesh - Food Prices, is a dataset provided by World Food Programme (WFP).

This dataset contains Food Prices data for Bangladesh. Food prices data comes from the World Food Programme and covers foods such as maize, rice, beans, fish, and sugar for 76 countries and some 1,500 markets. It is updated weekly but contains to a large extent monthly data. The data goes back as far as 1992 for a few countries, although many countries started reporting from 2003 or thereafter.

5.1.1.1 Timeline of the dataset

The Dataset covers prices starting from **Jan 15, 2004** to **Mar 15, 2019.**

This dataset updates Every week and was last Updated on April 28, 2019

5.1.1.2 Size of the dataset

The dataset had 3412 datas in total.

5.1.1.3 Covered items

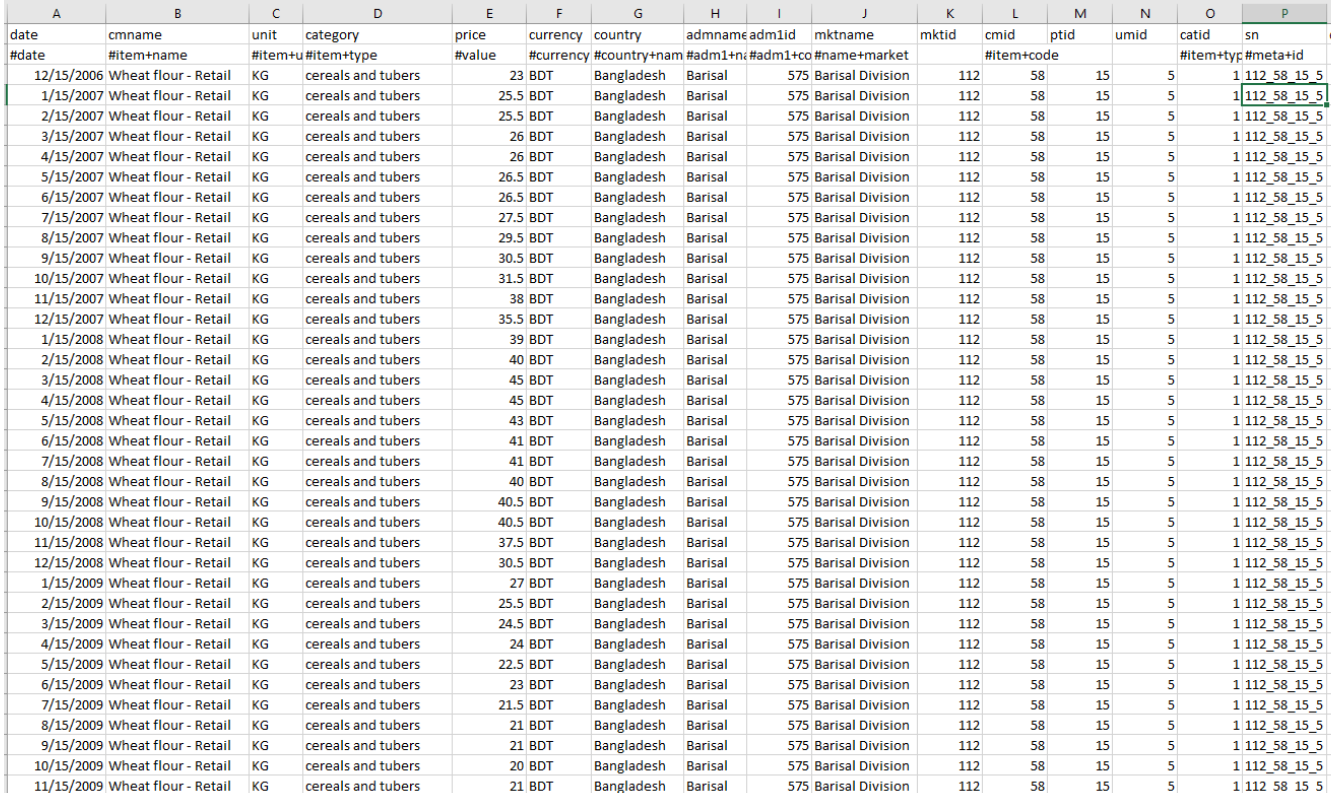
The main category of items and the corresponding commodities of those categories are-

- cereals and tubers: wheat flour and rice(coarse)
- oil and fats: palm oil and
- pulses and nuts: Lentils (masur)

5.1.1.4 Covered divisions

i. Dhaka
ii. Chittagong
iii. Khulna
iv. Barisal
v. Rajshahi
vi. Sylhet

.



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
date	cmname	unit	category	price	currency	country	admname	adm1id	mktname	mktid	cmid	ptid	umid	catid	sn
#date	#item+name	#item+u	#item+type	#value	#currency	#country+nam	#adm1+na	#adm1+co	#name+market		#item+code			#item+typ	#meta+id
12/15/2006	Wheat flour - Retail	KG	cereals and tubers	23	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
1/15/2007	Wheat flour - Retail	KG	cereals and tubers	25.5	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
2/15/2007	Wheat flour - Retail	KG	cereals and tubers	25.5	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
3/15/2007	Wheat flour - Retail	KG	cereals and tubers	26	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
4/15/2007	Wheat flour - Retail	KG	cereals and tubers	26	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
5/15/2007	Wheat flour - Retail	KG	cereals and tubers	26.5	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
6/15/2007	Wheat flour - Retail	KG	cereals and tubers	26.5	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
7/15/2007	Wheat flour - Retail	KG	cereals and tubers	27.5	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
8/15/2007	Wheat flour - Retail	KG	cereals and tubers	29.5	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
9/15/2007	Wheat flour - Retail	KG	cereals and tubers	30.5	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
10/15/2007	Wheat flour - Retail	KG	cereals and tubers	31.5	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
11/15/2007	Wheat flour - Retail	KG	cereals and tubers	38	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
12/15/2007	Wheat flour - Retail	KG	cereals and tubers	35.5	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
1/15/2008	Wheat flour - Retail	KG	cereals and tubers	39	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
2/15/2008	Wheat flour - Retail	KG	cereals and tubers	40	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
3/15/2008	Wheat flour - Retail	KG	cereals and tubers	45	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
4/15/2008	Wheat flour - Retail	KG	cereals and tubers	45	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
5/15/2008	Wheat flour - Retail	KG	cereals and tubers	43	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
6/15/2008	Wheat flour - Retail	KG	cereals and tubers	41	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
7/15/2008	Wheat flour - Retail	KG	cereals and tubers	41	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
8/15/2008	Wheat flour - Retail	KG	cereals and tubers	40	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
9/15/2008	Wheat flour - Retail	KG	cereals and tubers	40.5	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
10/15/2008	Wheat flour - Retail	KG	cereals and tubers	40.5	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
11/15/2008	Wheat flour - Retail	KG	cereals and tubers	37.5	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
12/15/2008	Wheat flour - Retail	KG	cereals and tubers	30.5	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
1/15/2009	Wheat flour - Retail	KG	cereals and tubers	27	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
2/15/2009	Wheat flour - Retail	KG	cereals and tubers	25.5	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
3/15/2009	Wheat flour - Retail	KG	cereals and tubers	24.5	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
4/15/2009	Wheat flour - Retail	KG	cereals and tubers	24	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
5/15/2009	Wheat flour - Retail	KG	cereals and tubers	22.5	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
6/15/2009	Wheat flour - Retail	KG	cereals and tubers	23	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
7/15/2009	Wheat flour - Retail	KG	cereals and tubers	21.5	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
8/15/2009	Wheat flour - Retail	KG	cereals and tubers	21	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
9/15/2009	Wheat flour - Retail	KG	cereals and tubers	21	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
10/15/2009	Wheat flour - Retail	KG	cereals and tubers	20	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5
11/15/2009	Wheat flour - Retail	KG	cereals and tubers	21	BDT	Bangladesh	Barisal	575	Barisal Division	112	58	15	5	1	112_58_15_5

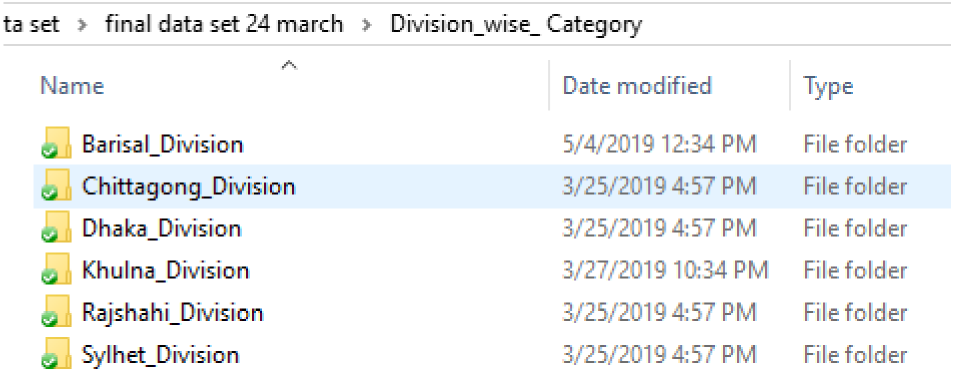
5.1.2 Pre-processing the dataset

5.1.2.1 Cleaning the dataset

The dataset was cleaned to discard the unnecessary columns such as – Unit, Currency, Country, market id, commodity id and category id along with meta id. These attributes were redundant in a sense that only 1 value of this attributes were present. It would have been an added burden of features to the machine learning algorithms.

5.1.2.2 Division-wise categorizing

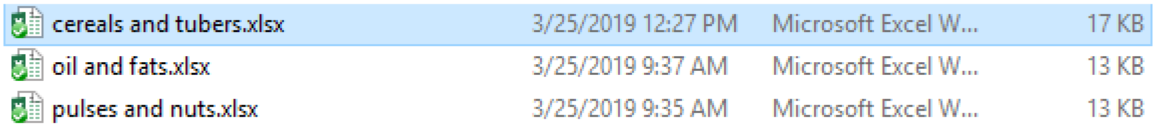
The commodity prices varied from a division to another. So the dataset was grouped according to division.



ta set › final data set 24 march › Division_wise_ Category

Name	Date modified	Type
Barisal_Division	5/4/2019 12:34 PM	File folder
Chittagong_Division	3/25/2019 4:57 PM	File folder
Dhaka_Division	3/25/2019 4:57 PM	File folder
Khulna_Division	3/27/2019 10:34 PM	File folder
Rajshahi_Division	3/25/2019 4:57 PM	File folder
Sylhet_Division	3/25/2019 4:57 PM	File folder

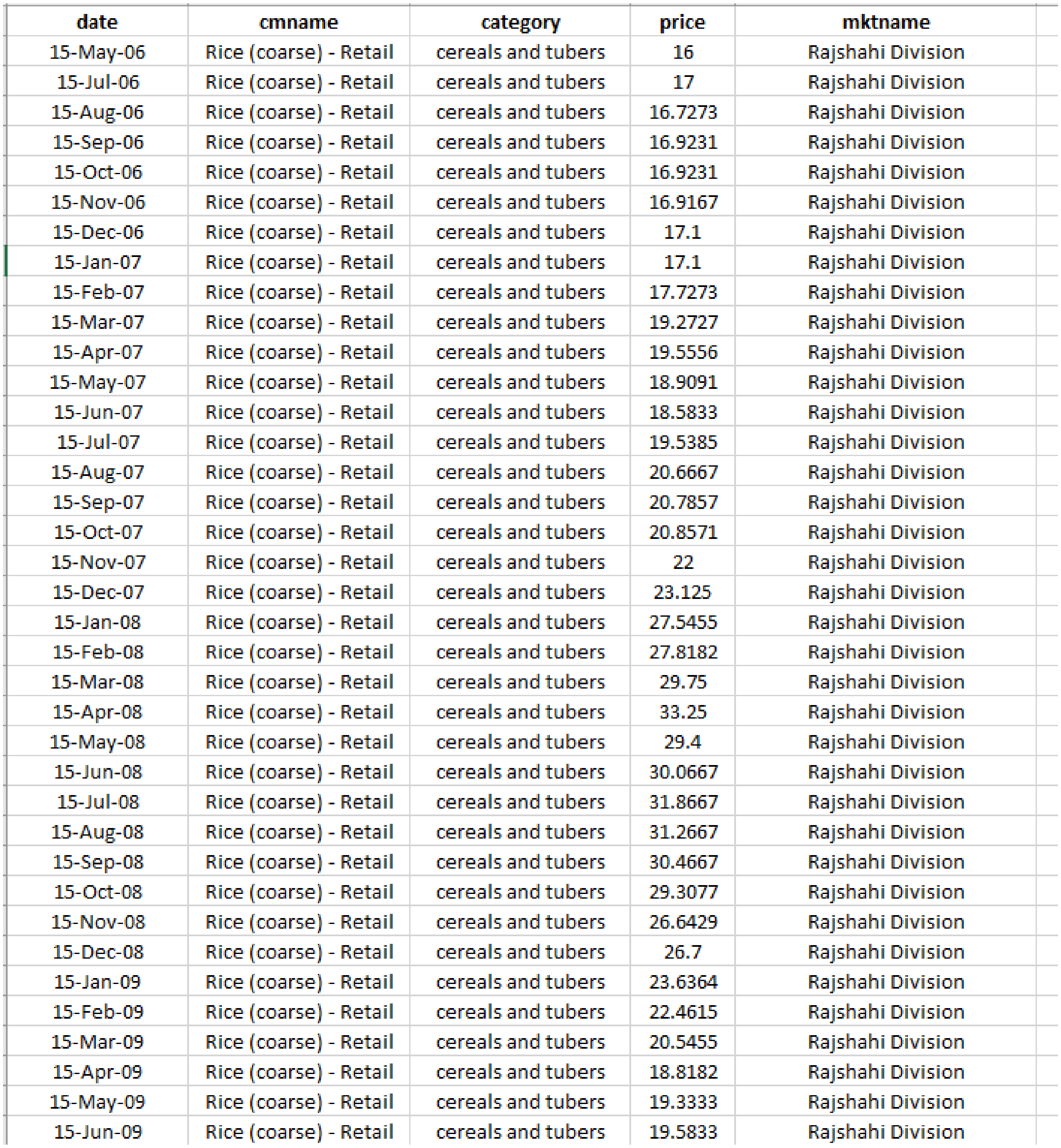
Then the dataset was further divided according to commodity category since the price of one commodity had less or no impact on the other's price. The work concentrated on rice price under Rajshahi division.



cereals and tubers.xlsx	3/25/2019 12:27 PM	Microsoft Excel W...	17 KB
oil and fats.xlsx	3/25/2019 9:37 AM	Microsoft Excel W...	13 KB
pulses and nuts.xlsx	3/25/2019 9:35 AM	Microsoft Excel W...	13 KB

After preprocessing, 146 rows remained under Rice price of Rajshahi Division.

A snapshot is given below –



date	cmname	category	price	mktname
15-May-06	Rice (coarse) - Retail	cereals and tubers	16	Rajshahi Division
15-Jul-06	Rice (coarse) - Retail	cereals and tubers	17	Rajshahi Division
15-Aug-06	Rice (coarse) - Retail	cereals and tubers	16.7273	Rajshahi Division
15-Sep-06	Rice (coarse) - Retail	cereals and tubers	16.9231	Rajshahi Division
15-Oct-06	Rice (coarse) - Retail	cereals and tubers	16.9231	Rajshahi Division
15-Nov-06	Rice (coarse) - Retail	cereals and tubers	16.9167	Rajshahi Division
15-Dec-06	Rice (coarse) - Retail	cereals and tubers	17.1	Rajshahi Division
15-Jan-07	Rice (coarse) - Retail	cereals and tubers	17.1	Rajshahi Division
15-Feb-07	Rice (coarse) - Retail	cereals and tubers	17.7273	Rajshahi Division
15-Mar-07	Rice (coarse) - Retail	cereals and tubers	19.2727	Rajshahi Division
15-Apr-07	Rice (coarse) - Retail	cereals and tubers	19.5556	Rajshahi Division
15-May-07	Rice (coarse) - Retail	cereals and tubers	18.9091	Rajshahi Division
15-Jun-07	Rice (coarse) - Retail	cereals and tubers	18.5833	Rajshahi Division
15-Jul-07	Rice (coarse) - Retail	cereals and tubers	19.5385	Rajshahi Division
15-Aug-07	Rice (coarse) - Retail	cereals and tubers	20.6667	Rajshahi Division
15-Sep-07	Rice (coarse) - Retail	cereals and tubers	20.7857	Rajshahi Division
15-Oct-07	Rice (coarse) - Retail	cereals and tubers	20.8571	Rajshahi Division
15-Nov-07	Rice (coarse) - Retail	cereals and tubers	22	Rajshahi Division
15-Dec-07	Rice (coarse) - Retail	cereals and tubers	23.125	Rajshahi Division
15-Jan-08	Rice (coarse) - Retail	cereals and tubers	27.5455	Rajshahi Division
15-Feb-08	Rice (coarse) - Retail	cereals and tubers	27.8182	Rajshahi Division
15-Mar-08	Rice (coarse) - Retail	cereals and tubers	29.75	Rajshahi Division
15-Apr-08	Rice (coarse) - Retail	cereals and tubers	33.25	Rajshahi Division
15-May-08	Rice (coarse) - Retail	cereals and tubers	29.4	Rajshahi Division
15-Jun-08	Rice (coarse) - Retail	cereals and tubers	30.0667	Rajshahi Division
15-Jul-08	Rice (coarse) - Retail	cereals and tubers	31.8667	Rajshahi Division
15-Aug-08	Rice (coarse) - Retail	cereals and tubers	31.2667	Rajshahi Division
15-Sep-08	Rice (coarse) - Retail	cereals and tubers	30.4667	Rajshahi Division
15-Oct-08	Rice (coarse) - Retail	cereals and tubers	29.3077	Rajshahi Division
15-Nov-08	Rice (coarse) - Retail	cereals and tubers	26.6429	Rajshahi Division
15-Dec-08	Rice (coarse) - Retail	cereals and tubers	26.7	Rajshahi Division
15-Jan-09	Rice (coarse) - Retail	cereals and tubers	23.6364	Rajshahi Division
15-Feb-09	Rice (coarse) - Retail	cereals and tubers	22.4615	Rajshahi Division
15-Mar-09	Rice (coarse) - Retail	cereals and tubers	20.5455	Rajshahi Division
15-Apr-09	Rice (coarse) - Retail	cereals and tubers	18.8182	Rajshahi Division
15-May-09	Rice (coarse) - Retail	cereals and tubers	19.3333	Rajshahi Division
15-Jun-09	Rice (coarse) - Retail	cereals and tubers	19.5833	Rajshahi Division

We split the dataset into training and test data, Training data consists of beginning of survey to 2018 september with 146 records and Test data consists of 2018 september afterwards (March 2015) data with 7 records.

5.2 Model buildup

We built three models on the **WEKA timeseries environment.**

Weka (>= 3.7.3) now has a dedicated time series analysis environment that allows forecasting models to be developed, evaluated and visualized.

Weka's time series framework takes a machine learning/data mining approach to modeling time series by transforming the data into a form that standard propositional learning algorithms can process. It does this by removing the temporal ordering of individual input examples by encoding the time dependency via additional input fields. These fields are sometimes referred to as "lagged" variables. Various other fields are also computed automatically to allow the algorithms to model trends and seasonality. After the data has been transformed, any of Weka's regression algorithms can be applied to learn a model.

5.2.1 Target selection

At the top left of the basic configuration panel is an area that allows the user to select which target field(s) in the data they wish to forecast. The system can jointly model multiple target fields simultaneously in order to capture dependencies between them. Because of this, modeling several series simultaneously can give different results for each series than modeling them individually. When there is only a single target in the data then the system selects it automatically. In the situation where there are potentially multiple targets the user must select them manually.

5.2.2 Basic parameters

At the top right of the basic configuration panel is an area with several simple parameters that control the behavior of the forecasting algorithm.

Number of time units

The first, and most important of these, is the **Number of time units** *to forecast* text box. This controls how many time steps into the future the forecaster will produce predictions for. The default is set to 1, i.e. the system will make a single 1-step-ahead prediction. For the airline data we set this to 24 (to make monthly predictions into the future for a two year period) and for the wine data we set it to 12 (to make monthly predictions into the future for a one year period). The units correspond to the periodicity of the data (if known). For example, with data recorded on a daily basis the time units are days.

Time stamp

Next is the **Time stamp** drop-down box. This allows the user to select which, if any, field in the data holds the time stamp. If there is a date field in the data then the system selects this automatically. If there is no date present in the data then the "<Use an artificial time index>"

option is selected automatically. The user may select the time stamp manually; and will need to do so if the time stamp is a non-date numeric field (because the system can't distinguish this from a potential target field). The user also has the option of selecting "<None>" from the drop-down box in order to tell the system that no time stamp (artificial or otherwise) is to be used.

Periodicity

Underneath the Time stamp drop-down box is a drop-down box that allows the user to specify the **Periodicity** of the data. If a date field has been selected as the time stamp, then the system can use heuristics to automatically detect the periodicity - "<Detect automatically>" will be set as the default if the system has found and set a date attribute as the time stamp initially. If the time stamp is not a date, then the user can explicitly tell the system what the periodicity is or select "<Unknown>" if it is not known. Periodicity is used to set reasonable defaults for the creation of *lagged variables* (covered below in the **Advanced Configuration** section). In the case where the time stamp is a date, Periodicity is also used to create a default set of fields derived from the date. E.g. for a monthly periodicity, **month of the year** and **quarter** fields are automatically created.

Skip list

Below the Periodicity drop-down box is a field that allows the user to specify time periods that should not count as a time stamp increment with respect to the modeling, forecasting and visualization process. For example, consider daily trading data for a given stock. The market is closed for trading over the weekend and on public holidays, so these time periods do not count as an increment and the difference, for example, between market close on Friday and on the following Monday is one time unit (not three). The heuristic used to automatically detect periodicity can't cope with these "holes" in the data, so the user *must* specify a periodicity to use and supply the time periods that are not to considered as increments in the **Skip list** text field.

The Skip list field can accept strings such as "weekend", "sat", "tuesday", "mar" and "october", specific dates (with optional formatting string) such as "2011-07-04@yyyy-MM-dd", and integers (that get interpreted differently depending on the specified periodicity). For daily data an integer is interpreted as the day of the year; for hourly data it is the hour of the day and for monthly data it is the month of the year. For specific dates, the system has a default formatting string ("yyyy-MM-dd'T'HH:mm:ss") or the user can specify one to use by suffixing the date with "@<format>". If all dates in the list have the same format, then it only has to be specified once (for the first date present in the list) and then this will become the default format for subsequent dates in the list.

Confidence intervals

Below the Time stamp drop-down box is a check box and text field that the user can opt to have the system compute confidence bounds on the predictions that it makes. The default confidence level is 95%. The system uses predictions made for the known target values in the training data to set the confidence bounds. So, a 95% confidence level means that 95% of the true target values fell within the interval. Note that the confidence intervals are computed for each step-ahead level independently, i.e. all the one-step-ahead predictions on the training data are used to compute the one-step-ahead confidence interval, all the two-step-ahead predictions are used to compute the two-step-ahead interval, and so on.

Perform evaluation

By default, the system is set up to learn the forecasting model and generate a forecast beyond the end of the training data. Selecting the **Perform evaluation** check box tells the system to perform an evaluation of the forecaster using the training data. That is, once the forecaster has been trained on the data, it is then applied to make a forecast at each time point (in order) by stepping through the data. These predictions are collected and summarized, using various metrics, for each future time step forecasted, i.e. all the one-step-ahead predictions are collected and summarized, all the two-step-ahead predictions are collected and summarized, and so on. This allows the user to see, to a certain degree, how forecasts further out in time compare to those closer in time. The *Advanced Configuration* panel allows the user to fine tune configuration by selecting which metrics to compute and whether to hold-out some data from the end of the training data as a separate test set

5.2.3 Output

Output generated by settings available from the basic configuration panel includes the training evaluation (shown in the previous screenshot), graphs of forecasted values beyond the end of the training data (as shown in Section 3.1), forecasted values in text form and a textual description of the model learned. There are more options for output available in the *advanced configuration panel* (discussed in the next section). The next screenshot shows the model learned on the airline data. By default, the time series environment is configured to learn a linear model, that is, a linear support vector machine to be precise. Full control over the underlying model learned and its parameters is available in the advanced configuration panel.

5.2.4 Advanced Configuration

The advanced configuration panel gives the user full control over a number of aspects of the forecasting analysis. These include the choice of underlying model and parameters, creation of lagged variables, creation of variables derived from a date time stamp, specification of "overlay" data, evaluation options and control over what output is created. Each of these has a dedicated sub-panel in the advanced configuration and is discussed in the following sections.

Base learner

The **Base learner** panel provides control over which Weka learning algorithm is used to model the time series. It also allows the user to configure parameters specific to the learning algorithm selected. By default, the analysis environment is configured to use a linear support vector machine for regression (Weka's SMOreg). This can easily be changed by pressing the Choose button and selecting another algorithm capable of predicting a numeric quantity.

Lag creation

The **Lag creation** panel allows the user to control and manipulate how lagged variables are created. Lagged variables are the main mechanism by which the relationship between past and current values of a series can be captured by propositional learning algorithms. They create a "window" or "snapshot" over a time period. Essentially, the number of lagged variables created determines the size of the window. The basic configuration panel uses the **Periodicity** setting to set reasonable default values for the number of lagged variables (and hence the window size) created. For example, if you had monthly sales data then including lags up to 12 time steps into the past would make sense; for hourly data, you might want lags up to 24 time steps or perhaps 12.

The left-hand side of the lag creation panel has an area called *lag length* that contains controls for setting and fine-tuning lag lengths. At the top of this area there is a **Adjust for variance** check box which allows the user to opt to have the system compensate for variance in the data. It does this by taking the log of each target before creating lagged variables and building the model. This can be useful if the variance (how much the data jumps around) increases or decreases over the course of time. Adjusting for variance may, or may not, improve performance. It is best to experiment and see if it helps for the data/parameter selection combination at hand. Below the adjust for variance check box is a **Use custom lag lengths** check box. This allows the user to alter the default lag lengths that are set by the basic configuration panel. Note that the numbers shown for the lengths are not necessarily the defaults that will be used. If the user has selected "<Detect automatically>" in the periodicity drop-down box on the basic configuration panel then the actual default lag lengths get set when the data gets analysed at run time. The **Minimum lag** text field allows the user to specify the minimum previous time step to create a lagged field for - e.g. a value of 1 means that a lagged variable will be created that holds target values at time - 1. The **Maximum lag** text field specifies the maximum previous time step to create a lagged variable for - e.g. a value of 12 means that a lagged variable will be created that holds target values at time - 12. All time periods between the minimum and maximum lag will be turned into lagged variables. It is possible to fine tune the creation of variables within the minimum and maximum by entering a range in the **Fine tune lag selection** text field. In the screenshot below we have weekly data so have opted to set minimum and maximum lags to 1 and 52 respectively. Within this we have opted to only create lags 1-26 and 52.

Periodic attributes

The **Periodic attributes** panel allows the user to customize which date-derived periodic attributes are created. This functionality is only available if the data contains a date time stamp. If the time stamp is a date, then certain defaults (as determined by the Periodicity setting from the basic configuration panel) are automatically set. For example, if the data has a monthly time interval then *month of the year* and *quarter* are automatically included as variables in the data. The user can select the **customize** checkbox in the *date-derived periodic creation* area to disable, select and create new custom date-derived variables.

Evaluation

The Evaluation panel allows the user to select which evaluation metrics they wish to see, and configure whether to evaluate using the training data and/or a set of data held out from the end of the training data. Selecting Perform evaluation in the Basic configuration panel is equivalent to selecting Evaluate on training here. By default, the mean absolute error (MAE) and root mean square error (RMSE) of the predictions are computed. The user can select which metrics to compute in the *Metrics* area in on the left-hand side of the panel. The available metrics are:

i. Mean absolute error (MAE): sum(abs(predicted - actual)) / N
ii. Mean squared error (MSE): sum((predicted - actual)^2) / N
iii. Root mean squared error (RMSE): sqrt(sum((predicted - actual)^2) / N)
iv. Mean absolute percentage error (MAPE): sum(abs((predicted - actual) / actual)) / N
v. Direction accuracy (DAC): count(sign(actual_current - actual_previous) == sign(pred_current - pred_previous)) / N
vi. Relative absolute error (RAE): sum(abs(predicted - actual)) / sum(abs(previous_target - actual))
vii. Root relative squared error (RRSE): sqrt(sum((predicted - actual)^2) / N) / sqrt(sum(previous_target - actual)^2) / N)

The relative measures give an indication of how the well forecaster's predictions are doing compared to just using the last known target value as the prediction. They are expressed as a percentage, and lower values indicate that the forecasted values are better predictions than just using the last known target value. A score of >=100 indicates that the forecaster is doing no better (or even worse) than predicting the last known target value. Note that the last known target value is relative to the step at which the forecast is being made - e.g. a 12-step-ahead prediction is compared relative to using the target value 12 time steps prior as the prediction (since this is the last "known" actual target value).

The text field to the right of the **Evaluate on held out training** check box allows the user to select how much of the training data to hold out from the end of the series in order to form an independent test set. The number entered here can either indicate an absolute number of rows, or can be a fraction of the training data (expressed as a number between 0 and 1).

5.3 Model training

5.3.1 Gaussian Regression

We developed a Gaussian regression model with the following configuration:

Scheme	GaussianProcesses -L 1.0 -N 0 -K "PolyKernel -E 1.0 -C 250007" -S 1
Lagged and derived variable options	-F price -L 1 -M 12 -month -quarter
price	Gaussian Processes
Kernel used:	Linear Kernel: K(x,y) = <x,y>

All values shown based on Normalize training data

5.3.2 SMOReg

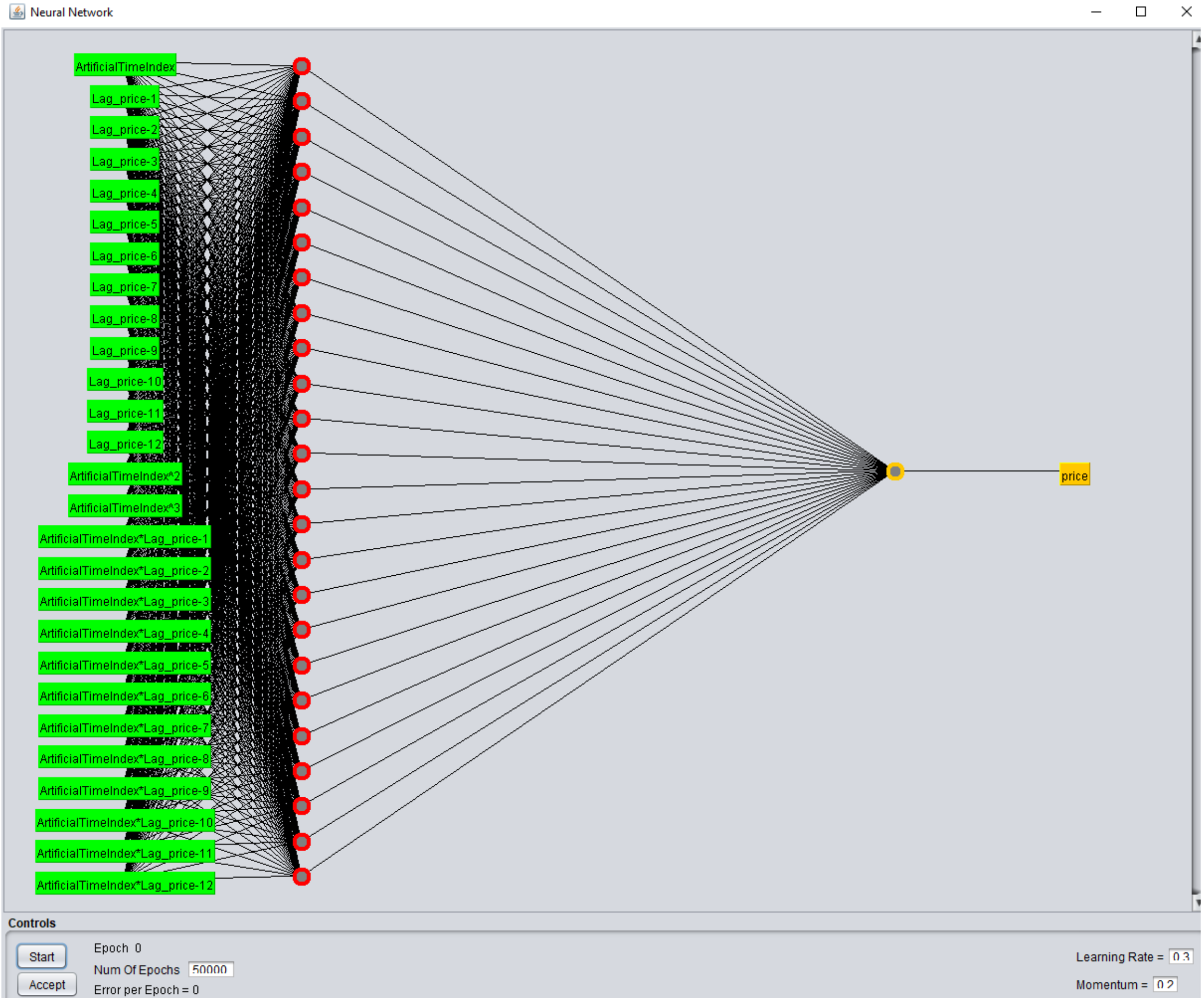
We developed a Support Vector Machine Regressor SMOreg model with the following configuration:

Scheme	SMOreg -C 1.0 -N 0 -I "RegSMOImproved -T 0.001 -V -P 1.0E-12 -L 0.001 -W 1" -K "PolyKernel -E 1.0 -C 250007"
Lagged and derived variable options	-F price -L 1 -M 12 -month -quarter
price	ArtificialTimeIndex and Lag_price-1 to Lag_price-12
Kernel used:	PolyKernel
Number of kernel evaluations	10731 (98.651% cached)

5.3.3 MLP

We developed a Multi-Layer Perceptron model of the ANN family with the following configuration:

Scheme	MultilayerPerceptron -L 0.3 -M 0.2 -N 50000 -V 0 -S 0 -E 20 -H 24 -G -R
Lagged and derived variable options	-F price -L 1 -M 12 -month -quarter
price	ArtificialTimeIndex and Lag_price-1 to Lag_price-12
Number of Hidden Layers	20
Activation function	Sigmoid
Learning Rate and Momentum	0.3 and 0.2
Number of iteration	50000


Neural Network
ArtificialTimeIndex
Lag_price-1
Lag_price-2
Lag_price-3
Lag_price-4
Lag_price-5
Lag_price-6
Lag_price-7
Lag_price-8
Lag_price-9
Lag_price-10
Lag_price-11
Lag_price-12
ArtificialTimeIndex^2
ArtificialTimeIndex^3
ArtificialTimeIndex*Lag_price-1
ArtificialTimeIndex*Lag_price-2
ArtificialTimeIndex*Lag_price-3
ArtificialTimeIndex*Lag_price-4
ArtificialTimeIndex*Lag_price-5
ArtificialTimeIndex*Lag_price-6
ArtificialTimeIndex*Lag_price-7
ArtificialTimeIndex*Lag_price-8
ArtificialTimeIndex*Lag_price-9
ArtificialTimeIndex*Lag_price-10
ArtificialTimeIndex*Lag_price-11
ArtificialTimeIndex*Lag_price-12
price
Controls
Start
Accept

Epoch 0
Num Of Epochs 50000
Error per Epoch = 0
Learning Rate = 0.3
Momentum = 0.2

5.4 Model Validation Measures

5.4.1 Mean Absolute Deviation (MAD)

The mean absolute deviation (MAD) is the sum of absolute differences between the actual value and the forecast divided by the number of observations.

5.4.2 Mean Square Error (MSE)

The most commonly used error metric. It penalizes larger errors because squaring larger numbers has a greater impact than squaring smaller numbers. The MSE is the sum of the squared errors divided by the number of observations.

5.4.3 Root Mean Square Error (RMSE)

The square root of the MSE.

5.4.4 Mean Absolute Percentage Error (MAPE)

The average of absolute errors divided by actual observation values.

Chapter 6

Result

6.1 Result and Accuracy of model 1

When using GP, we are able to add prior knowledge and specifications about the shape of the model by selecting different kernel functions. Here PolyKernel was used.

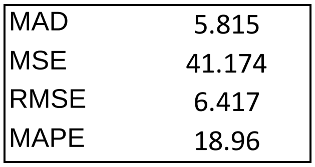
For model 1, the performance metrics are as follows-

MAD	4.835
MSE	29.798
RMSE	5.459
MAPE	14.91

6.2 Result and Accuracy of model 2

SVM are motivated through statistical learning theory. The theory characterizes the performance of learning machines using bounds on their ability to predict future data. Training many local SVMs instead of a single global one can lead to significant improvement in the performance of a learning machine.

For model 2, the performance metrics are as follows

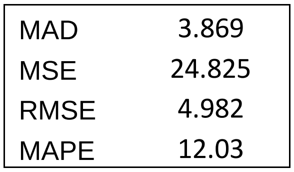


MAD	5.815
MSE	41.174
RMSE	6.417
MAPE	18.96

6.3 Result and Accuracy of model 3

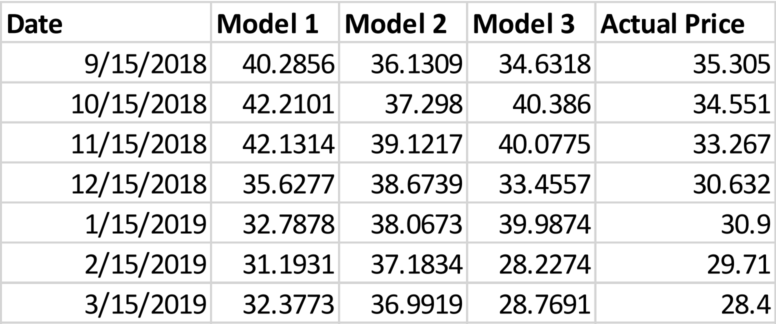
Neural networks are able to capture the underlying pattern or autocorrelation structure within a time series even when the underlying law governing the system is unknown or too complex to describe.

For model 3, the performance metrics are as follows

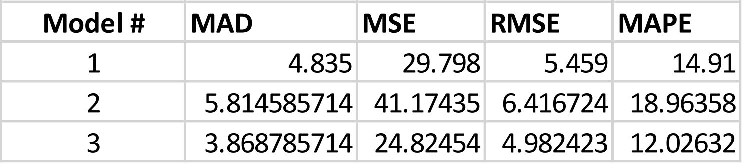


MAD	3.869
MSE	24.825
RMSE	4.982
MAPE	12.03

6.4 Comparison



Date	Model 1	Model 2	Model 3	Actual Price
9/15/2018	40.2856	36.1309	34.6318	35.305
10/15/2018	42.2101	37.298	40.386	34.551
11/15/2018	42.1314	39.1217	40.0775	33.267
12/15/2018	35.6277	38.6739	33.4557	30.632
1/15/2019	32.7878	38.0673	39.9874	30.9
2/15/2019	31.1931	37.1834	28.2274	29.71
3/15/2019	32.3773	36.9919	28.7691	28.4



Model #	MAD	MSE	RMSE	MAPE
1	4.835	29.798	5.459	14.91
2	5.814585714	41.17435	6.416724	18.96358
3	3.868785714	24.82454	4.982423	12.02632

It is evident from results that MLP(model 3) is better among the worst.

Chapter 7

Conclusion

Gaussian process (GP) directly captures the model uncertainty. As an example, in regression, GP directly gives you a distribution for the prediction value, rather than just one value as the prediction. This uncertainty is not directly captured in neural networks

ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Modern neural networks are non-linear statistical data modelling tools. They are usually used to model complex relationships between inputs and outputs or to find patterns in data.

Challenges faced

The lack of data was troublesome as machine learning approaches are data hungry.

ANN is in a sense the ultimate ‘black boxes’. Apart from defining the general architecture of a network and perhaps initially seeding it with a random number, the user has no other role than to feed it input and watch it train and await the output. In fact, it has been said that with ANN, “you almost don't know what you're doing”. The final product of this activity is a trained network that provides no equations or coefficients defining a relationship (as in regression) beyond its own internal mathematics. The network 'IS' the final equation of the relationship.

The research is constrained by the WEKA workbench and the techniques it makes available as part of its suite of offering. WEKA was originally selected as the main tool of analysis because it was familiar to the researcher but this of course meant that the techniques used were limited only to those available in the WEKA suite. It is reasonable to believe more popular tools such the R programming language or deep learning software could also provide interesting results.

Future working scope

Division wise commodities price can be predicted to detect trends in data.

Even though much research has been done on regression, two important aims seem to have directed these studies. One is the fitness of the estimation or accuracy; the other is the interpretability of the constructed model. New research can be conducted in these traditional directions. Additionally, research can also be directed to increase the efficiency, both in computational complexity and storage. Also, constructing systems that deal with outliers, noise, missing values and irrelevant features is very important, since databases today have a very large number of records and attributes. Models that enables the detection and interpretation of interactions between attributes can also be researched in the future.

References

[1] Bangladesh Bureau of Statistics (BBS) Yearbook of Agricultural Statistics–2014. 26 (January, 2016): 3.

[2] Trenca, Ioan & ZOICAS-IENCIU, Adrian. (2010). The correlation between the market risk and the liquidity risk in the romanian banking sector. *Annals of the University of Oradea : Economic Science. 1.*

[3] T. Gubanova & L.Lohr & T.Park(April 18–19, 2005). Forecasting Organic Food Prices: Emerging Methods for Testing and Evaluating Conditional Predictive Ability.*NCR-134 Conference on Applied Commodity Price Analysis, Forecasting, and Market Risk Management,St. Louis, Missouri.*

[4] B.Guha & G. Bandyopadhyay (March,2016). Gold Price Forecasting Using ARIMA Model. *Journal of Advanced Management Science Vol. 4, No. 2.*

[5] J.Wang & Y. Chena &X. Wang & X. Zheng & J. Zhao (2010). Cycle Phase Identification and Factors Influencing the Agricultural Commodity Price Cycle in China: Evidence from Cereal Prices *International Conference on Agricultural Risk and Food Security.*

[6] G.Li & S. Xu & Z. Li (2010) Short-Term Price Forecasting For Agro-products Using Artificial Neural Networks. *International Conference on Agricultural Risk and Food Security 2010*

[7] T.R. Cook & A.S. Hall (September 2017). Macroeconomic Indicator Forecasting with Deep Neural Networks. *Federal Reserve Bank of Kansas City.*

[8] L.Nuno. Stock Market Price Prediction Using Linear and Polynomial Regression Models. *University of New Mexico Computer Science Department.* Retrieved from Web

[9] R.Kumar & A.Balara (2014). Time Series Forecasting Of Nifty Stock Market Using Weka.*JRPS International Journal for Research Publication & Seminar Vol 05 Issue 02 March -July 2014*

[10] J.Harris. A machine learning approach to Forecasting consumer food prices (2017). *Dalhousie University Halifax, Nova Scotia.* Retrieved from Web

[11] AchtertE, BöhmC, Kriegel H-P, Kröger P (2005).Online hierarchical clustering in a data warehouse environment. In: *Proceedings of the 5th international conference on data mining (ICDM), Houston, TX, pp 10–17*

[11] P. Hall, J. Dean, I.K. Kabul, J. Silva, "An Overview of Machine Learning with SAS Enterprise Miner," SAS Institute Inc, 2014. Retrieved May 6, 2015 from http://support.sas.com/resources/papers/proceedings14/SAS313-2014.pdf.

[12]. L. Breiman, "Random Forests," Machine Learning, vol. 45, 2001, 5–32.

[13]. C.P. Igiri, O.U. Anyama, A.I. Silas, I. Sam, "A Comparative Analysis of K-NN and ANN Techniques in Machine Learning,"

International Journal of Engineering Research and Technology (IJERT), vol. 4(3), 2015, 420–425.

[14] I. Uysal & H.A.Guvenir (1999). An overview of regression techniques for knowledge discovery. *The Knowledge Engineering Review.*

[15] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A Training Algorithm for Optimal Margin Classifiers. In D. Haussler, editor, *5th Annual ACM Workshop on COLT, pages 144-152, Pittsburgh, PA, 1992.* ACM Press.

[16] T.Evgeniou & M.Potil (2001). Support Vector Machines: Theory and Applications.

[17] R. Hecht-Nielsen, *Neurocomputing (Addison-Wesley, Reading, MA, 1990).*

[18] S.K. Shevade, S.S. Keerthi, C. Bhattacharyya, K.R.K. Murthy: Improvements to the SMO Algorithm for SVM Regression. In*: IEEE Transactions on Neural Networks, 1999.*