MACHINE LEARNING BASED ANALYSIS AND PREDICTION OF CROP YIELD AND PRICES OF AMAN, AUS AND BORO RICE

By
Ruposri Bhattacharjee
18321060
Kazi Andelib Mamun
17121017
Kazi Saad Asif
16321081
Shaian Khan
17321031

A thesis submitted to the Department of Electrical and Electronic Engineering in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering

Electrical and Electronic Engineering
BRAC UNIVERSITY
September, 2021

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at BRAC University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Ruposri Bhattacharjee
18321060

Kazi Andelib Mamun
17121017

Kazi Saad Asif
16321081

Shaian Khan
17321031

Approval

The thesis/project titled "Machine Learning Based Analysis and Prediction of Crop Yield and Prices of Aman, Aus and Boro Rice" submitted by

1. Ruposri Bhattacharjee (18321060)
2. Kazi Andelib Mamun (17121017)
3. Kazi Saad Asif (16321081)
4. Shaian Khan(17321031)

of Summer, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Bachelor of Science in Electrical and Electronic Engineering on October 3rd, 2021.

Examining Committee:

Supervisor:
(Member)

Abu S.M. Mohsin, PhD
Associate Professor, Department of Electrical and Electronic Engineering
BRAC University

Program Coordinator:
(Member)

Abu S.M. Mohsin, PhD
Associate Professor, Department of Electrical and Electronic Engineering
BRAC University

Departmental Head:
(Chair)

Md. Mosaddequr Rahman, PhD
Professor and Chairperson, Department of Electrical and Electronic Engineering
BRAC University

Abstract

Agriculture has been the driving force of the Bangladesh economy. In the agricultural sector, farmers are largely incapable of using scientific technology to maximize crop yield and identify which crops can be grown in specific weather and soil conditions. Recently, the effectiveness of machine learning-based algorithms in utilizing large datasets to accurately predict and provide descriptive solutions holds promising potential in solving this problem by giving descriptive farming advice and fertilizer usage for farmers and proper yield predictions for better import and export policies. Therefore, this paper aims to use historical weather and climate data (such as temperature, rainfall, average bright sunshine, cloud coverage, etc.) and agricultural data such as fertilizer, soil type, and soil moisture to provide predictions on the yield of Aus, Boro, and Aman that can be expected to grow in a region as well as predict the future rice prices of Dhaka depending on existing data. After analysis it was found that there is direct correlation of high accuracy between weather factors such as average rainfall, average minimum temperature, average maximum temperature, average yearly temperature, average bright sunshine, average cloud coverage, relative humidity, average wind speed, latitude, longitude and altitude and yearly yield of Aus, Aman and Boro rice when algorithms such as KNN, linear regression, random forest, and XGBoost were implemented. Furthermore, correlation was found among soil type, soil moisture, fertilizer type and crop yield. Finally, a price prediction of three different types of rice –Aus, Aman, and Boro – between Dhaka and Delhi was conducted using models such as ARIMA and SARIMAX.

Keywords: Aus, Aman, Boro, Yield Prediction, Machine Learning, and Agriculture, KNN, Linear Regression, XGBoost, Random Forest, ARIMA, SARIMAX.

Acknowledgement

This is to express our utmost gratitude and humble recognition of the role Dr. A S M Mohsin has played in the inception, guidance, and careful curation of the development of this thesis. Without his in-depth knowledge of Machine Learning, its application capabilities and data procurement this work would probably not have come to fruition. In our experience, we have had the unique pleasure of having a mentor who has not only been a guiding light but a supporting and patient leader for a team of fresh aspiring researchers. Apart from our thesis advisor, we would also like to express gratitude to our respective parents for nurturing us and aiding us in enabling an education for us that has presented us with this unique opportunity to work on a novel and impactful thesis project.

We would also like to thank Nahid Hossain Taz for his help.

Table of Contents

List of Tables

Chapter -3

Chapter -4

Chapter -5

List of Figures

Chapter 2

Chapter 3

Chapter 4

Chapter 5

List of Acronyms

Al	Artificial Intelligence
DL	Deep Learning
RelU	Rectified linear Unit
AF	Activation Function
ANNs	Artificial Neural Networks
ARIMA	AutoRegressive Integrated Moving Average
KNN	K-Nearest Neighbors algorithm
XGboost	eXtreme Gradient Boost
MSE	Mean Squared Error
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
MPE	Mean Percentage Error
MAPE	Mean Absolute Percentage Error

Chapter 1

1.1 Introduction

Machine learning is about to play a very important role in human lives in the near future. In this project, we are using Machine learning which includes Artificial Neural Networks (ANN). ANN is a process in which an algorithm is written such that it imitates the way human brains procure data, convert it into knowledge about a pattern, and learn to recognize it. In machine learning, we use neural networks with three or more layers: the input layer, the hidden layer(s), and output layer. In our project, we are using machine learning to find maximum crop yield production of three different types of rice, namely Aus, Boro, and Aman, by using parameters such as weather, area, soil, fertilizer type, and year. Bangladesh has immensely fertile lands which allow considerable potential in crop yield enhancement. [1] Our project focuses on predicting crop yield to better inform the government about the upcoming supply of major staple crops, fertilizer suggestions for farmers to ensure higher yields, and district-wise crop price prediction for a more stable market for consumers. A robust machine learning-based platform helps in generating predictive information about the yield rate of crops, the correlation of the crop yield between weather factors such as temperature and rainfall as well as helps to predict the market prices of crops using various machine learning algorithms or techniques. [1] Moreover, it also allows better control over the supply and demand for these crops, and also the dependency on foreign imports will be less due to this. Additionally, this can be impactful in empowering farmers with proper knowledge about cultivation and usage of fertilizers in a more controlled manner to ensure the better yield of a crop as well as creates transparency among farmers and consumers about the prices of the crops and thus a more transparent marketplace will be ensured. [2] In this process, we are using 44 years of crop yield data, 65 years of weather and agricultural data, and crop prices across districts of Bangladesh using statistical associative models to predict crop yield, suggest better soil and area-specific fertilizer usage and prices. We plan on training the data first, validate it using another portion of the available datasets, and include an input portal in the platform for future data to be fed into the model for further development and higher accuracy in the future. All these processes, simulation, data collection, building statistical model, collecting and comparing data, generating yield prediction, fertilizer suggestion, and estimated prices will be achieved via machine learning-based neural networks through a web application. Since Bangladesh is heavily dependent on agriculture for sustenance as well as economic growth, this robust and multi-faceted platform will help to create more informed policymakers, market regulators, and empowered farmers. [3]

1.2 Literature Review

Most of the research on Machine Learning and Agriculture across the world is regarding crop yield prediction based on weather data, soil data, and geospatial imagery. Some research is also done based on fertilizer prediction based on maximum yield. It has been done from manually recorded soil data over the years from Government Statistics Offices, satellite imagery, and climate data which have been digitized and published online. [4] And crop yield data of 43 years has been taken from the Bangladesh Bureau of Statistics. [5] Algorithms such as Support Vector and Random Forest have been used to suggest crop fertilizers(Mr. Santosh Mahagoanka, 2019). [6] Crop prediction using parameters like humidity, rainfall, and soil type of a specific area like Karnataka has been done using K-NN algorithms. (H. K Karthikeya, et al, 2020) [7] Classification of crop productivity has been done using K-NN as well. In Nigeria, data mining has been utilized for the recommendation of fertilizers using SVM and XGboost with high accuracy. (Terungwa Simon Yange, et al, 2020).[8] Integrated Application of Remote Sensing and GIS has also been used previously to forecast Aman rice production in Bangladesh.[9] In this thesis, we have used a custom dataset and applied Machine Learning models like ARIMA, Linear Regression, Random Forest, K-Nearest Neighbor, and XGBoost to predict prices, yield, and Fertilizer for Aman, Boro, and Aus rice.

1.3 Motivation: Machine Learning

Artificial intelligence (AI) will play a large part in lives in the future and machine learning is an impactful AI tool. Humans invent and create solutions to make their daily lives easier. Machine learning will play a crucial role in pivoting our standard of living in the upcoming years - in sectors such as medical, finance, traffic systems, trade, business, and many others. Machine learning is helping to calculate and predict the dynamic systems in the world. In our project, we are using three or more neural networks to train, validate and test the models. Using machine learning models, we can train computers to do a specific job fast and effectively. Bangladesh is a developing country but it is rich in many natural resources. One of the richest resources of the country is fertile soil. Given this fertile soil, a large number of people seek to make a living as farmers. As Bangladesh is rich in the agricultural sector, we aim to use machine learning so that we increase the production of our staple crops, as well as distribute and regulate their prices more effectively. This will help to meet the basic needs of the people and provide a means to make income from our agricultural sectors. If we can predict the supply of major staple crops like Aus, Aman, and Boro rice accurately, we can improve our import policies; with a more robust fertilizer suggestive system farmers can better cultivate and the transparent market price will make way for a fairer and more integrated market place across the country.

1.4 Background of Machine Learning

Machine learning employs mathematical and logical algorithms to evaluate and train data to make predictions or analyses to aid with better decisions in the future. [10] With the advent of Machine Learning, now it is possible for computer software to communicate with each other and humans, autonomously drive cars, predict natural disasters, etc. Machine learning is an Artificial Intelligence(AI) that enables machines to ascertain and amend naturally without being explicitly programmed.[10] It mainly focuses on model implementation by using different algorithms and allows computer systems to access data and train themselves to provide better results.[10] In the 1950s, the concept of machine learning came into the picture when Alan Turing published an article answering the question of whether machines can think or not. [11] He proposed a hypothesis which was called the Turing test.[11] Frank Rosenblatt designed the first neural network for computers in 1957 which is commonly called the perception model and this perception algorithm was designed to classify visual inputs and categorize subjects.[11] In 1967, the nearest neighbor algorithm was written that allowed computer systems to use pattern recognition.[11] During the 1990s, the knowledge-driven approach of machine learning was shifted to a data-driven approach.[10]

The applications of machine learning play a vital role by identifying key factors in achieving maximum crop yield at minimum cost.[10] It is found in recent studies that various estimation techniques were published which were simple and accurate, to identify whether artificial neural network models could predict crop yield effectively, evaluate model performance and compare the effectiveness of various models.[12] Crop yield prediction and crop selection, smart irrigation systems, weather forecasting are some of the agricultural sectors of machine learning applications.[13] To ensure the maximum crop yield, it is important to select an appropriate crop that depends on various factors like soil, climate, crop yield, market price, the geography of the region, fertility level, etc.[12] Additionally, Machine Learning techniques like Artificial Neural networks, K-Nearest Neighbor, Linear Regression, Decision Trees have been effectively used in the agricultural sectors to minimize losses and make models more accurate.[14]

1.5 Aim/ Objective of the project

Our proposed system aims at predicting the crop yield based on the weather factor and also agricultural factors which impact the yield of the crop such as fertilizer, land and soil type, soil moisture, etc of three different types of paddy-Aus, Aman, and Boro and we also aim to predict rice prices of Dhaka city by exploring the previous year’s data. In the agricultural sector of

Bangladesh, farmers face different types of problems that we want to address and analyze to provide some solutions through our research. Some of the problems include lack of accurate prediction of the yield of different types of crops such as Aus, Aman, Boro, lack of accurate fertilizer usage for farmers, Inability of policymakers to predetermine prices of rice, and lack of transparency in crop prices across districts in real-time, etc. Using Machine Learning techniques, the system constructs a predictive model by considering different factors such as maximum temperature, average rainfall, minimum temperature, relative humidity, average wind speed, average bright sunshine, and cloud coverage. The modus operandi of this project is to analyze how rainfall and climate change historically correlates with the growth of crops, how different fertilizer usage can affect the yield, and test how changes in fertilization can affect the yield. For this purpose, we have used different machine learning techniques for example Linear regression, Random forest, K-Nearest Neighbor, Extreme gradient boosting algorithm to design, develop and implement the model, and the overall performance of the implemented model is basically assessed by the predicted accuracy. Moreover, we will suggest which fertilizer, land type, and soil type has a great impact on each rice yield and compare the results to find out which algorithm is best in terms of predicting the yield of Aus, Aman, and Boro as well as we will be predicting the near future rice prices of these crops in Dhaka city in Bangladesh.

1.6 Thesis structure

- In chapter 2, we have given a brief explanation of machine learning and the conceptual overview of artificial neural networks which includes the architecture of ANN, basic components, and the training procedure of machine learning projects using various performance metrics is also included in this chapter.
- In chapter 3, we have focused on the different algorithms of Machine Learning for instance Linear regression, K-Nearest Neighbor, Extreme gradient boosting, and Random forest, to predict the model of crop yield using various fertilizer, land types, soil types including soil moisture, etc. as well as we have compared among the algorithms to interpret our result and detailed description of model implementation is also included.
- In chapter 4, we again implemented those four algorithms to predict the crop yield using various weather parameters such as minimum and maximum temperature, rainfall, humidity, etc. of Aus, Aman, and Boro and showed the correlation or impact of the parameters in the yield prediction.
- In chapter 5, by using the ARIMA model we have predicted rice prices in the near future as well as compared our result and model with another country.
- Chapter 6 concludes the paper with remarks, limitations of our paper, and the future scope of our research.

Chapter 2
Conceptual Overview

2.1 Architecture of ANN

ANN is also known as the Artificial Neural Network that consists of artificial neurons.[3] Let us define the ANN. The process in which a set of algorithms or code is used to integrate more than one layer to process a set of data like the human brain is known as ANN. [3] ANN is mostly used to mimic the function of the human brain as it is most efficient to solve the most complex calculation and problems. [3] There are many applications of ANN. Some of them involve: [3]

- biological systems,
- real-time adaptability,
- possibility of asynchronous processing,
- multidirectional execution,
- high level of parallelism,
- developed mathematical foundations. [3]

Artificial Neural Network (ANN) is a computing technology that imitates the biological neurological structure of the human brain.[15] They consist of units or nodes called neurons. They are as follows: Convolutional Neural Networks, Generative Adversarial Networks, Radial Basis Networks, and Multi-Layered Perceptrons.[15]

ANN is the most complex system or process which is used in modeling the performance of fuel cell systems (Milewski and Świrski, 2009; Świrski and Milewski, 2009). [15] The technology is arranged in such a way that it emulates a behavior without an algorithm-based solution bypassing available experimental data. ANN is used for both training and modeling cell behavior. The error back-propagation algorithm is utilized. [15]

2.1.1 Artificial Neural Network Components - Input Layers, Neurons, and Weights [16]

Simple Figure:

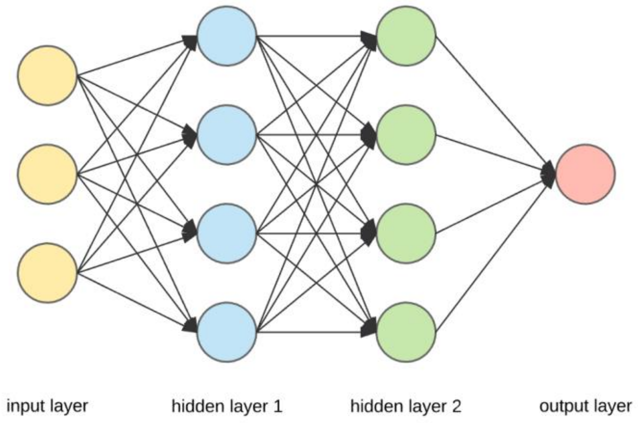


Figure:2.1 Simple ANNs [17]

Complex Figure:

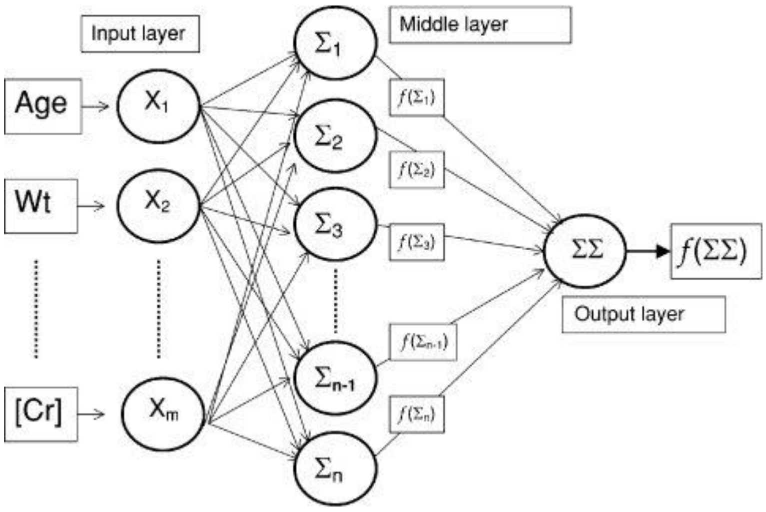


Figure:2.2 Complex Formula of ANNs [3]

This figure is showing how the ANNs work. Here there is an input layer in which a data set is given like age, number, etc. [16] After the input layer, Middle is used to process and integrate data with weight and this layer can be more than one layer. [16] Finally, there is an output layer that sums all the results of the last layer of the middle layer. Hence the output layer gives us the final result. [16]

2.1.2 Artificial neural networks: Algorithm

The artificial neural networks process is done in some steps by the algorithm which is vital for working properly. [17] The initiated steps of ANN are divided the data into three different sets:

1. **Training dataset –** It explains the weights between nodes to the Neural Network. [17]
2. **Validation dataset –** It fine-tunes the performance of the Neural Network. [17]
3. **Test dataset –** It determines the accuracy and margin of error of the Neural Network. [17]

These initiated steps of the ANN process help us to write the code or algorithm in a well-defined manner that the ANNs mimic the function of the human brain. [16] Once the data is segmented into these three steps and then ANNs algorithms are applied to them for training the Neural Network.[17] The procedure used for facilitating the training process in ANNs is known as optimization while the algorithm used is known as optimizer.[17] There are multiple optimization algorithms available, each with different characteristics and aspects such as processing speed, memory requirements, and numerical precision. [17]

2.2 Basic ANN Components

The architecture of ANN is formed on the function of the Neural Network which is similar to the neurons of the human brain.[18] Neuron is the primary segment of the human brain which is responsible for communication with other neurons in the brain to detect what is happening throughout the body and in turn generate reactive responses.[18] The main job of the artificial neural network is to mimic this behavior.[18] An artificial neural network contains some basic processing elements or components which include: [16]

- Layers
- Weights
- Bias
- Activation function.

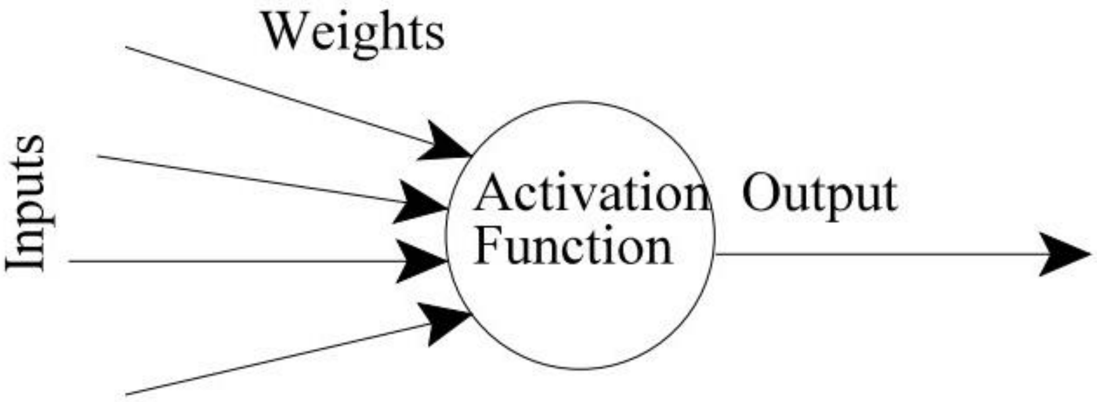


Figure:2.3 Primary elements of Artificial Neural Network [19]

An artificial neural network consists of neurons that are assembled in various layers like input layer, output layer and also hidden layer.[16] Moreover, there is a popular neural network system called feed-forward neural network that contains an input layer to execute pattern recognition by receiving data from an external source, a hidden layer that separates the other layers and features of input which passed through this hidden layer from the input layer part and an output layer that identifies what the object is.[19] In the case of ANN, it uses a training algorithm called backpropagation to learn the datasets, and the weights of neurons are modified depending on the error rates between the target output and the substantial output.[18]
The basic structure of neural networks looks like this:

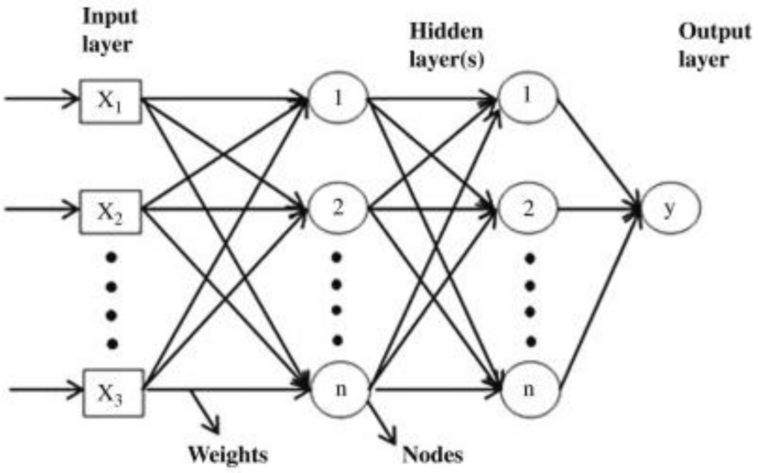


Figure:2.4 Basic Structure of Artificial neural network [18]

2.2.1 Input Layer

The input layer is the primary layer to process the workflow of the artificial neural network and input information is received by this input layer in the form of numbers, texts, and image pixels, etc. The neural network contains an input layer that is composed of artificial input neurons and the system takes the initial data from the input layer which is further processed by other subsequent layers. [16] The other components of artificial neural networks such as the hidden layer or output layer may have convolutional or encoding layers. [16]

One of the marked characteristics is that as the input layer is the primary layer of the network, it does not take any information from previous layers and thus the input layer is composed of passive neurons. Moreover, artificial neurons may sometimes have a set of weighted neurons which function on the basis of weights. Basically, input layers do the job of measuring features. [18]

2.2.2 Hidden Layer

As soon as the features pass through the neurons, they will be transferred to the hidden layer from the input layer and each neuron will do some kind of processing. In the case of artificial neural networks, hidden layers exist between input and output layers where weighted inputs are being taken by the neurons and it produces output through an activation function. [16] Hidden layers in neural networks are structured in different ways such as in some cases weights are assigned randomly to the input when there might be some other cases where the weighted input has to go through a process called backpropagation. [20] Hidden layers work like neurons in the brain in which it takes an input signal, processes them, and converts it to an output signal with the help of activation function. Inside of a hidden layer, two types of operations take place in which there will be the summation of each assigned weight multiplied by each input, and after that, an activation function such as sigmoid will be applied in the result. [16]

2.2.3 Output Layer

In an ANN, the output layer is the last layer of the system which produces the desired output. Output layers are built using different ways as it is responsible for the improvement of the end result. After passing the output of a particular neuron from the hidden layer to the output layer, there will be another weight assigned to that result and that result will also go through the process of applying the activation function. [16]

To understand the work of a neural network, we need to study all three layers that are the input layer, hidden layers, and output layer.

2.2.4 Weights & Bias

In an artificial neural network, there are some weights randomly assigned to the input which helps to produce the desired output in the system. These weighted inputs in a neuron engage in different procedures in which they go through an activation function and display the output or decision. [18] In the backpropagation system, these weighted inputs can be modified or updated according to the output. Additionally, in the output layer, we will have our predicted output which needs to be compared with the actual output using the loss function and if this predicted output does not match with the actual output then these weights adjusting must be done in such a way that the actual and predicted output become same as well as it reduces the value of the loss function. [16] Thus, weights play an important role in neural network systems.

Bias is also a very important factor in neural networks similar to weights. Each neuron has a bias and the values which are assigned in biases are learnable just like weights. Specifically, bias determines whether a neuron is activated or not. Moreover, the addition of these biases increases the flexibility of the model to fit the available data. [16] To use the bias in any system, what needs to be done is pass the weighted sum plus the bias term to the activation function which helps to make the model more flexible and this bias also helps to shift the threshold as per our desired value, instead of passing the weighted sum directly to the activation function.

2.2.5 Activation Function

An activation function is critical in any neural network system. There are multiple activation functions such as Sigmoid, ReLU, and Leaky ReLU activation functions. [21]

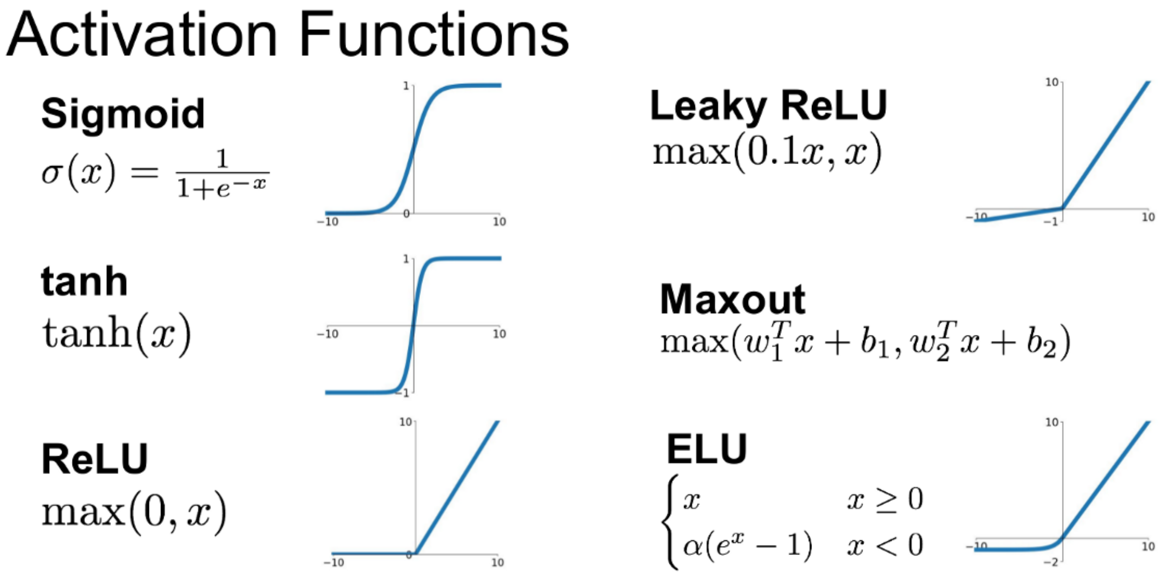


Figure:2.5 Activation function of ANN [21]

Sigmoid:

A sigmoid activation function is basically used in Logistic regression. Inside a neural network, Weights are being multiplied by input features plus bias, and that term will go through the sigmoid activation function. [22] Once it is transferred to the activation function, this will transform the value between 0 to 1. In the sigmoid function, 0.5 is considered as the threshold, and what sigmoid does is, it can transform any value between 0 to 1 whether it is positive or negative. If the transformed value is below 0.5 then it is considered as 0 and if it is above 0.5 then it is considered as 1. Moreover, if the output is 1, it basically indicates that the neuron is activated and if the output is 0 that indicates the neuron is not activated and if the neuron is activated then it will transfer the signal and it will help in classifying the final output. [21]

One major disadvantage of the sigmoid function is that the output function is not zero-centered and time-consuming. [21]

ReLU:

Similar to sigmoid, weights are multiplied by input features plus bias and that function is passed to the ReLU activation function. If the function is passed to the ReLU activation function then the simple formula max(y,0) is getting applied where y basically indicates the output of weights multiplied by input features plus bias. [21]

If the value of ‘y’ is negative, then the output will always be 0 and if the output of ‘y’ is positive then the max of output will be a positive value or y. [21]

Whenever we try to find out the derivative of y using backpropagation, for all the values of y that are greater than 0, the derivative will be 1 and when the value of y is less than 0, the derivative will be 0. [21]

This method of this activation function is much more popular than the sigmoid as it is faster as well as there is no problem of gradient saturation problem in the ReLU activation function and it is mostly used to solve regression problems. [22]

In the case of the classification problem, the ReLU activation function can be used in the middle layer but in the final output layer, we always need to apply the sigmoid function.

But there is a problem in the ReLU activation function which is, it gets saturated at the negative region which means the output of the negative region becomes zero. [21]

Leaky ReLU:

We use Leaky ReLU in order to prevent the problem of the ReLU activation function. In Leaky ReLU, a constant will be multiplied with y and it will be set to 0.01y instead of 0. [21], [22] So, if we find the derivative of the negative region, we will get a smaller value instead of getting 0 which can solve the dying ReLU problem because whenever we have negative values due to negative weights and we do the derivative, the value will be 0.01 which is greater than 0. [22]

However, in actual operation, it is not fully proved yet that Leaky ReLU outperformed ReLU.

2.3 Training of Artificial Neural Network (ANN)

In the late ’80s and early ’90s, neural networks failed to significantly outperform the successful combination of the Hidden Markov model (HMM) with acoustic models based on Gaussian mixtures. [3] Nowadays, the neural network has outperformed most of the previous models. Behind this outperformance of Neural Networking, there are 4 major factors.[3] These are

1. A larger number of input and output units greatly improve their performance.
2. Faster hardware makes it possible to train deep neural networks effectively.
3. Initializing the weight balance.
4. Deep Neural Networks

Neural Networking is a big process in all the sectors of human society. So, the question is “What is Neural Networking?”. The process in which a series of algorithms integrate to analyze underlying relationships of data set by mimicking the human brain operates is known as Artificial Neural Networking. It is done by training the Artificial Neural Networking. In the medical sector, the ANN is used to detect cancer, heart block, various infections, analyses the various viruses and bacteria, etc. [3] Other sectors like controlling traffic systems, internet servers, Agriculture production, etc. also are using ANNs process to make the complex system easy to run and administrate. [3]

In the modern age, agriculture needs higher performance and effective production of crops and vegetables as pollution of the world is increasing day by day. A neutral network process is also used in the agricultural sector so that we can produce more crops and vegetables in a very effective way in a small place and store them for a long period of time. Artificial neural networks are also known as ANN which is one of the most popular tools of ANNs for solving various classification and prediction tasks.

2.3.1 Back Propagation

Backpropagation is of the utmost importance and a basic component of neural networks. [20] Theoretically, backpropagation is based on the below mentioned fundamentals: [20]

The relation between sets of data is created through trial and error by the system.

1. The system is able to exhibit the type of internal representation that it has stabilized for the various tasks it has had to learn in order to carry out the multiplicity of tasks after having learned the sort of appropriate connections among its data sets via its structure. [20]

2. The system can easily operate generalizations and functions on other similar tasks ones it has already learned. [20]

3. The only memory of the system itself is the relations or the connection, which have become stabilized among the system’s data set during the learning of the several tasks. [20]

4. Most of the system data-sets are of a discriminant or sub-conceptual type. [20]

Based on this discriminant or sub-conceptual type, they can be of three logical types:

- Input data sets
- Output data sets
- Hidden data sets

5. The connection between the system data sets are uni-oriented. [20]

Besides, Backpropagation has 4 distinct systems that help to make ANN's process much more efficient. [20] These are:

- Forward pass
- Loss function
- Backward pass
- Weight updating

These systems help us to understand the necessary customization and update of ANNs. ANNs themselves are complex and without 4 distinct systems, it will be difficult to process the data sets. [20]

2.3.2 Loss function

Artificial Neural Networks are trained using the stochastic gradient descent optimization algorithm in which loss function is used to design the model. The loss function gives a lot of room to customize the neural networks and it helps define how exactly the output of the network is connected with the rest of the network. [23] Most basic and simplified perspective, the loss function (LF) can be defined in two parameters:

- Predicted Output
- True Output

Neural Network Loss Visualization: Neural Network Loss Visualization function essentially measures the poor performance of our model by comparison between the predicting model and the actual value it is supposed to output. [23] If Y predict is too deviant from the Y value, then the Loss value becomes very high. However, if both values are almost identical, then the Loss value becomes very low. Hence, we need to keep a loss function that can penalize a model effectively to perform a much better outcome while it is training on a dataset. [23] If the loss is too high, then this large value of the dataset propagates through the network while it's training and the weights change a little more than usual. If this loss value is small then the weights don't change that much since the network already does a good enough job.

Now let's look at the classification of the loss functions that work in the Artificial Neural Network. These are: [23]

1. Binary Classification
2. Multiclass Classification
3. Multilabel Classification

Besides, regression loss is an important part of the loss function of ANNs. In regression, the model tries to predict a continuous value of the given data set of the ANNs system. Some regression models include:

- House price prediction
- Person Age prediction

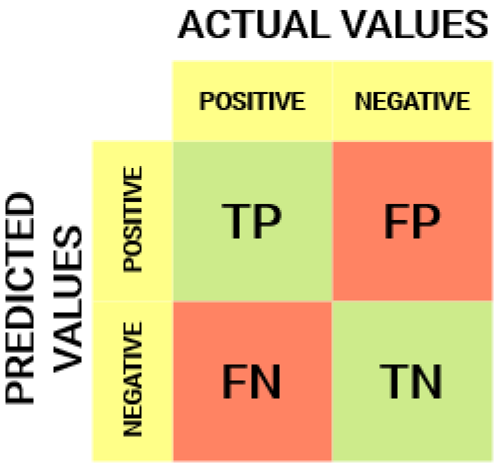
In regression models, the neural network needs to have at least one output node for each continuous value in which the systems are trying to predict. Performance of direct comparisons between the output value and the true value yields regression losses. [23] Mean squared error loss function is the most widely used loss function. Here, we simply compute the square of the difference between Y and Y predict and average all over the data. [23]

2.3.3 Calculation of Precision & Recall

Precision and Recall are other most important components of ANNs but to understand the calculation of Precision & Recall, first, we need to understand the confusion matrix and its characteristics. [24]

Confusion Matrix:

The matrix in which an N x N matrix is used for evaluating the performance of a classification model, where N is the number of target classes is known as the Confusion Matrix. [24]
For a binary classification problem, a 2 x 2 matrix is used with 4 values:



		ACTUAL VALUES	
		POSITIVE	NEGATIVE
PREDICTED VALUES	POSITIVE	TP	FP
	NEGATIVE	FN	TN

Figure:2.6 Confusion Matrix Truth table [24]

2.3.4 Calculation of Accuracy

Accuracy is also a part of the confusion matrix. With the help of it, we can accurately determine the accuracy of the model that lies between 0 to 100 percent. [24] The formula is

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$$ [24]

However, Accuracy is used in many sectors and branches so it is not a 100% percent correct way to evaluate any system.

2.3.5 Gradient descent

Gradient descent is an optimization algorithm used in training an ANNs model. The basis of it is a convex function and tweaks its parameters to minimize a given function or system to get the desired result. [25] We can actually define gradient descent as it is an algorithm which is also known as an optimization algorithm that is able to locate a minimum number of functions that are differentiable and this algorithm is used in the field of machine learning to find the values of variables that generally helps to minimize the cost function as much as possible. [25]
The equation below describes gradient descent:

$b = a - \gamma \nabla f(a)$ [25]

An example of a gradient descent 3D graph is given below:

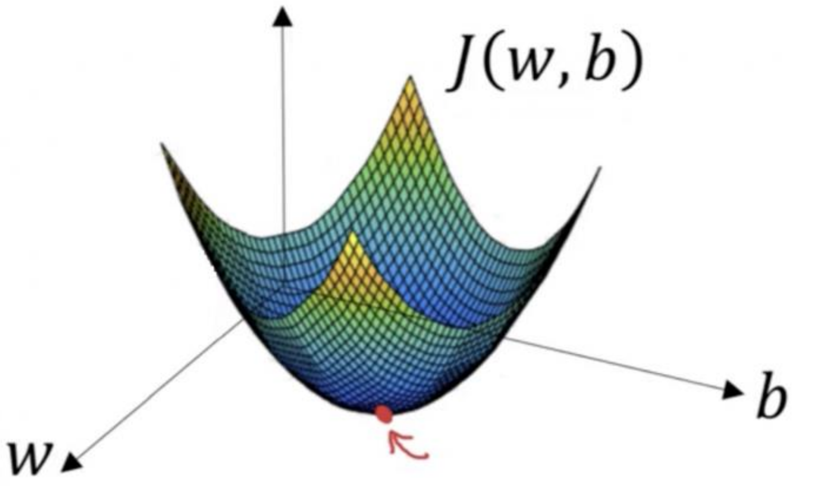


Figure:2.7 Gradient descent [25]

2.3.6 Learning Rate

Gradient descent is used to make ANNs with faster Learning rates. Learning rate is determined by global minimum which is aided by gradient descent. It determines the rate of movement towards the optimal weights.

The learning rate must be set to an appropriate value for gradient descent to attain the local minimum which is optimal. This is done mainly because if the steps are too big, the local minimum may not be achieved because it bounces to and from between the convex function of gradient descent (see left image below). If the learning rate is set to a very small value, gradient descent eventually reaches the local minimum but that may be prolonged. [25]

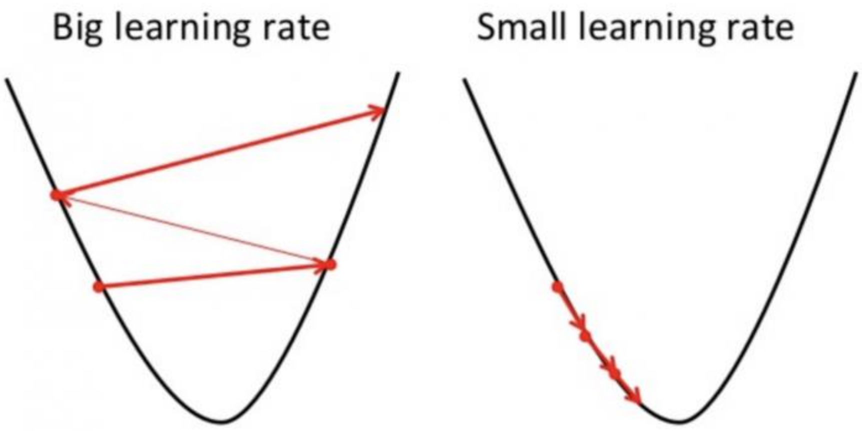


Figure:2.8 Learning rate [25]

2.3.7 Mini Batch SGD

Mini Batch SGD Is an important component of the ANNs process. The process in which a fluctuation of the gradient descent algorithm measures the error and updates the model for each case in the training dataset is known as Stochastic Gradient Descent. [26] It is also known as SGD. One of the formulas is given below:

for i in range (m):

$\theta_j = \theta_j - \alpha(\hat{y}^{i} - y^{i})x^{i}_{j}$ [26]

Now let us discuss mini-batch Stochastic Gradient Descent. [26] This process in which a variation of the gradient descent algorithm splits the training dataset into small batches that are used to calculate model error and update model coefficients is called mini-batch Stochastic Gradient Descent. [26] In Mini-Batch Stochastic Gradient Descent, the gradient is computed over mini-batches of training samples so that the parameters are updated and the loss function value (hopefully) continues to decrease over each iteration. [26] Advantages to Mini-Batch Stochastic Gradient Descent include:

- Model parameters are updated more frequently which allows for stronger convergence towards optimal parameter values. [26]
- Computationally more effective as Mini-Batch Stochastic Gradient Descent does not employ the full dataset. Using millions of training samples to update potentially thousands of parameters is extremely inefficient. [26]

2.4 Data Augmentation

Data augmentation is a technique normally used in ANNs to enlarge the dataset which is utilized in the training data and learning process. [27] The generation of new instances from the original data set is used to make the Data augmentation without changing the pattern of the data. [27] To train ANN models, usually, big data sets are needed by manual data collection or by taking already existing databases. [27] However, in few cases only a limited dataset or information on that topic is available. [27] Therefore, to get the required or desired dataset, we need to expand the size of the data set. [27] As a result data augmentation can be employed. [27]

In our project, we take the data of crops Aus, Aman, Boro, and their yield production, fertilizer as well as the price to calculate the prediction of production and price.

2.5 Algorithms Used

2.5.1 Linear Regression

Regression is a type of supervised learning technique which supports finding the correlation among variables. The output of the regression problem provides a real or continuous value. Linear regression is applied to estimate real values (prices of goods, sales made, cost of production, etc.) based on continuous variables or variables. A linear regression model with a single input variable is known as simple linear regression, and multiple linear regression if there are multiple input variables. It establishes the relationship between dependent and independent variables by fitting for the best fit line. This regression line or the best fit line is represented by the linear equation: [28], [29]

$$\mathbf{y = m * x + c}$$

In this equation:

- c - y-intercept
- m - slope
- x - independent variable
- y- dependent variable

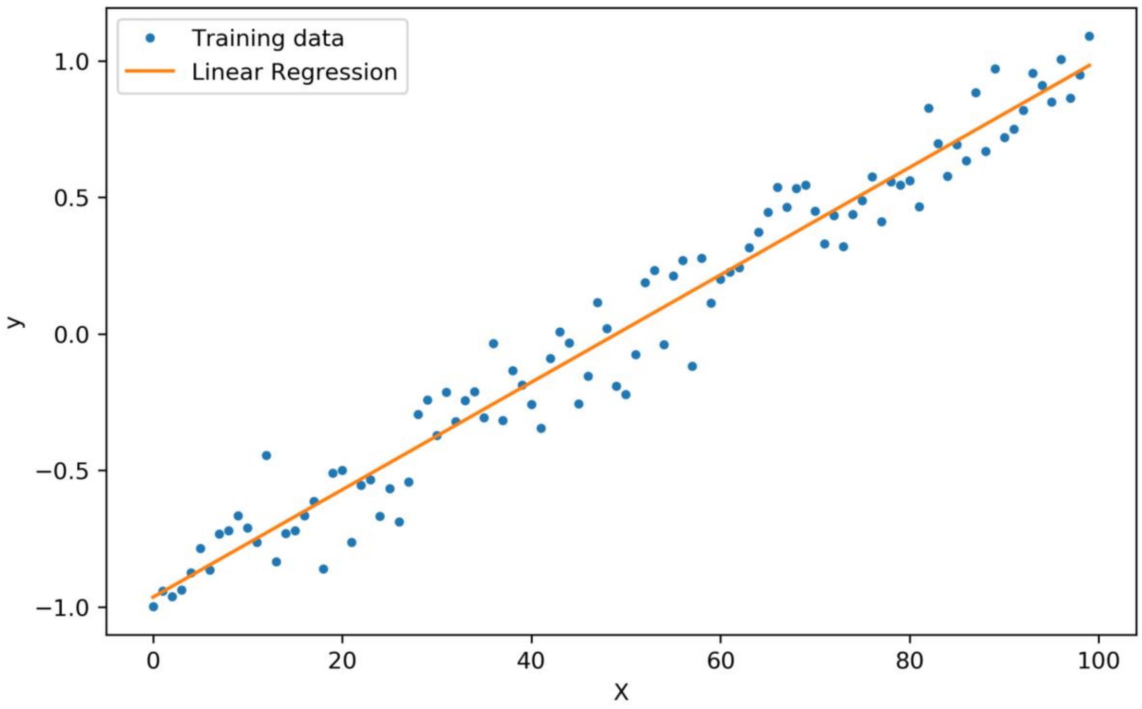


Figure:2.9 Linear Regression [29]

‘m’ and ‘c’ coefficients are derived based on the regression line and decreasing the sum of squared difference of distance between data points.

We have used a linear regression model with paddy yield for Aman, Aus, and Boro as the dependent variable and for the independent variables, we have chosen the annual averages for minimum and maximum temperature, rainfall, bright sunshine, relative humidity, wind speed, and cloud coverage as well as the latitude, longitude, and altitude of 15 districts in Bangladesh.

2.5.2 Random Forest Classification

Random Forest is an accumulation of decision trees. It consists of a supervised learning algorithm for predicting the output target features average by bootstrap aggregation or bagging (used for lowering variance error sources). The Random Forest model functions by utilizing a collection of decision trees known as “Forest” during its training phase. The decisions made by the majority of the trees are then selected by the Random Forest as the final result. The decision tree is a tree-shaped diagram whose branches represent a possible outcome. [28]

Random Forest work by: [28]

1) We assume the quantity of cases in a training set to be K. K case samples are taken randomly but an alternative value is placed. The sample is then used for the training set for growing the tree

2) If there are M numbers of input variables, at each node, a number m<<M is pointed out such that m numbers of variables are picked randomly out of the M and to split the nodes, the best split on this m is used. Moreover, during forest growing, the m value will remain constant.

3) There is no clipping so each tree is grown to the maximum extent possible.

4) Aggregating the predictions of the trees then predicts the new data.

Advantages of Random Forest

1) It can be used for both data regression and classification tasks. [30]

2) There is no overfitting of data as multiple decision trees are being used. This reduces the time it takes to train the model. [30]

3) It can estimate missing data. Therefore, it can maintain high accuracy when large portions of the data are missing [30]

4) It can handle larger datasets with higher dimensionality. Therefore, the accuracy is high. [30]

2.5.3 Autoregressive Integrated Moving Average Algorithm (ARIMA)

The model in which the observations are generated by the underlying process for a time series is known as ARIMA. The abbreviation of the word ARIMA is "Autoregressive Integrated Moving Average" . [31] This model works in 3 separate parts. They are as follows:

1. AR: Auto Regression
2. I: Integration
3. MA: Moving average

1. AR: Auto Regression:

This part utilizes the dependent relationships among observation and some number of lagged phenomena. It is known as Auto Regression. [31]
In the Auto-Regressive (AR only) model, Yt is dependent only on its own lags. The equation is
$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} + \epsilon_1$ [31]

2. I: Integration:

The integral is used in order to convert the time series into stationary by differentiating raw observations that are gained by subtracting again in observation from observation at the previous time step. [31]

3. MA: Moving average:

The Moving Average model deals with dependency of an observation on a residual error which is obtained from a moving average model applied to lag observations. [31] In a pure Moving Average (MA only) model, Yt depends only on the lagged forecast errors. The formula is:

$Y_t = \alpha + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + .. + \phi_q \epsilon_{t-q}$ [31]

Here, the errors of the Autoregressive models of the respective lags are represented as the error terms.

Now the final equation of AR and MA where the errors Et and E(t-1) are the errors from the following equations:

$Y_t = \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_0 Y_0 + \epsilon_t$ [31]

$Y_{t-1} = \beta_1 Y_{t-2} + \beta_2 Y_{t-3} + .. + \beta_0 Y_0 + \epsilon_{t-1}$ [31]

Now, these 3 models have individual parameters which are "p, d, q". These parameters are going to help us to substitute the integer values so that we can specify the ARIMA model. [31] The parameters of the ARIMA model are defined as follows:

- p: the number of lag observations of Lag order within the model. [31]
- d: The number of unique occurrences of difference or the degree of it. [31]
- q: The magnitude of moving average window of the order of it. [31]

In an ARIMA model, the time series was different at least once to make it stationary and the AR and the MA terms are merged together. [31] Therefore, the equation becomes:

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} +..+ \beta_p Y_{t-p} \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} +..+ \phi_q \epsilon_{t-q} \quad [31]$$

A linear regression model is made with the specific type and number of terms and the data is arranged in such a degree of difference so that it is at rest. This is done in order to withdraw trend and seasonal structures which negatively affect the regression model. Now a value like 0 also can be used as a parameter which shows us to not use the element in the model. [31] In this process, the ARIMA model can also be configured to perform as an ARMA model and furthermore, even a similar model as MA, AR, or I model. [31] Moreover, Adopting an ARIMA model helps to verify the presumption of the model in the raw observations and in the residual errors of forecasts from the model.

2.5.4 Extreme Gradient Boosted Algorithm:

The implementation of gradient boosted algorithms including decision trees that are devised for its fast speed and performance is known as XGBoost. [32] The word XGBoost stands for extreme Gradient Boosting. XGBoost is a software library and this library can be downloaded and installed on the machine as well as it has various interfaces to use. [32]

The main interfaces are given below: [32]

- Java and JVM languages like Scala and platforms like Hadoop
- Julia.
- R interface as well as a model in the caret package.
- Python interface, as well as a model in Scikit-learn.
- C++ (the language in which the library is written).
- Command Line Interface (CLI).

XGBoost Features

The library of XGBoost is based on performance and computational speed of models. [32] Besides, it does offer a number of advanced features which are given below:

❖ **Model Features**

The Scikit-learn implementation features are supported by the model. There are new additions like regularization. [32] The 3 main types of gradient boosting are:

- Gradient Boosting algorithm is otherwise known as gradient boosting machine or GBM and learning rate is included with its functionalities. [32]
- Stochastic Gradient Boosting with sub-sampling at the row, column and column per split levels. [32]
- Regularized Gradient Boosting uses the methods of regularization with L1 and L2. [32]

- **System Features**

A system for computing environment is given by the library which are:

- Parallelization is the process that consists of tree construction and it uses CPU core during training. [32]
- In order to train very large models, distributed computing is used. [32]
- Out-of-Core Computing is used for handling huge datasets that are difficult to insert into memory. [32]
- A data structure including algorithms is present in cache optimization which helps to get the best performance. [32]

- **Algorithm Features**

To get the best efficiency of computing time and memory resources, the implementation of algorithms is important. [32] Some features are used to train and implementation of the model which are given below:

- **Sparse Aware** is basically used for the implementation of missing data handling and it happens automatically. [32]
- Block Structure helps in the parallelization process by which tree construction is made. [32]
- By using Continued Training, it is possible to make necessary improvements in the existing fitted model on new data. [32]

XGBoost is free to use and an open-source software published under the Apache-2 license. [32]

2.5.5 K Nearest Neighbors

KNN or K nearest neighbor may be a machine learning method and calculation that can be utilized for both relapse and classification assignments. K nearest neighbor analyzes the name of a particular number of data points circumnavigating a wanted information point to form a forecast around the lesson that information point has a place to. For the fourth model, we have chosen K nearest neighbors Regressor as our machine learning model and have again generated mean absolute error, mean squared error, and root mean squared error metrics.

- **Process**
    - Number of K neighbors is selected.
    - Euclidean distance of K number of neighbors is calculated

- The nearest K neighbor is selected as per Euclidean distance
- Number of data points is counted among these K neighbors
- We assign the new data points to that category for which the number of the neighbor is maximum

❖ **K value**

- There is no fixed criterion for selecting k value.
- A random K value us initialized and computation begins
- A larger value of K yields more stable decision boundaries
- A plot is derived between error rate and K denoting values

❖ **Calculation of distance**

- Euclidean Distance: the sum of the squared differences between a new point (x) and an existing (y) is square rooted
- Manhattan Distance: the sum of the absolute differences is the distance between real vectors
- Hamming Distance: the sum of their absolute difference for distance between real vectors

❖ **Ways to perform KNN**

- Brute Force
- K-Dimensional Tree (Kd tree)
- Ball Tree

2.6 Metric for Performance Evaluation

2.6.1 Confusion Matrix

A tabular illustration that describes the performance of classification models on a set of test data for which the true values are known to us is known as the Confusion Matrix. [33] It is one of the most basic units for evaluating binary classifications. [33] In binary classification, we separate the data into the positive class and the negative class. Therefore, the confusion matrix is a 2 X 2 matrix, with the model predictions being in the columns and the true classes being in rows. [33]

True positives and true negatives are instances where the actual class is the same as the predicted class. [33]

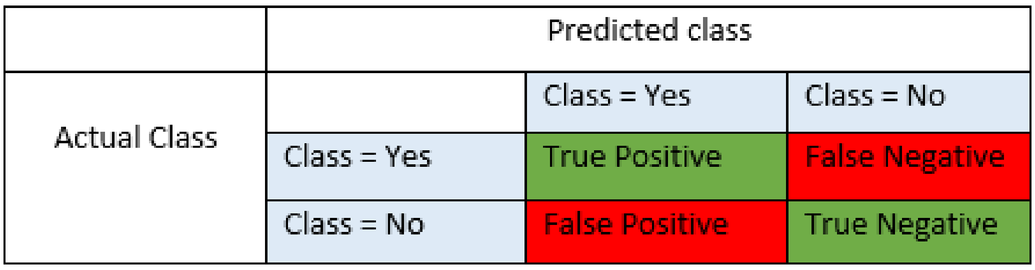
· **True Positives (TP)** are the values that were correctly positively predicted. This indicates that the actual class value is the same as the value of the predicted class in a positive sense. For example, if a model predicted it would rain on a given day, it did rain on that day. [33]

True Negatives (TN) are the values that were correctly negatively predicted. This indicates similarity between the value of the actual class and the value of the predicted class in a negative sense. For example, if a model predicted it would not rain on a given day and it did not rain on that day. [33]

False negatives and False positives are instances of contradiction between the actual class and the predicted class. [33]

- **False Positives (FP)** are values that were incorrectly positively predicted while the actual value is negative. For example, if the model predicted rain on a given day but there was no rain. [33]

- **False Negatives (FN)** are the values that were incorrectly predicted as negative values when the actual value is positive. For example, if the model predicted no rain on a given day when it in fact did rain.



	Predicted class		
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Figure:2.10 Confusion Matrix table [33]

2.6.2 Accuracy

The accuracy is the measure of the correct predictions the model makes. Accuracy is the mathematical ratio of correctly predicted observations of the model to total observations made by the model. However, accuracy is not always informative in situations where the data is imbalanced, for example, models that only predict positive or negative results. [33]

$Accuracy\ = \frac{TP\ +TN}{TP+FP+TN+FN}$ [33]

2.6.3 Precision

Precision sometimes referred to as the Positive Predicted Value (PPV) is normalized by everything that is predicted positive. The precision score points to the trait of the model to correctly predict the actual positives out of all the predictions made that were classed as positive. The precision score is used in instances of imbalanced classes to measure the success of prediction. Mathematically, it is the ratio of correctly predicted positive observations which are the True Positives to the total predicted positive i.e.; the sum of true positive and false-positive observations.

$Precision\ = \frac{TP}{TP+FP}$ [33]

2.6.4 Recall

Sensitivity or recall, the coverage, or the true positive rate of a model is normalized by the results that are actually positive. The recall score indicates the model's ability to correctly predict the positive values out of actual positives (true positives), that is it measures how efficient the model is at identifying all actual positives out of all positives that exist within a given dataset. Mathematically, recall is the ratio between correctly predicted positive observations and all observations in the actual positive class.

$Recall\ = \frac{TP}{TP+FN}$ [33]

2.6.5 F1 Score

The F1 score is the harmonic average of precision and recall which represents the model score as a function of recall score and precision. Therefore, it takes both false positives and false negatives into account.

$$F1\ Score\ = \frac{2*(Recall*Precision\ )}{Recall+\ Precision}$$ [33]

2.6.6 RMSE, MAE, MSE, r2_score

In order to evaluate the prediction error rates of the model and its performance in regression analysis, we check the Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and r2_score. Low values of MAE, MSE, and RMSE express the high accuracy of a regression model. High values of R square imply a better quality of model performance.

- **Mean Absolute Error (MAE):**

It represents the mean of the absolute difference between the actual values from the test data-set and predicted values obtained from the model. It calculates the mean of the remnants in the dataset.

$$M.A.E\ = \frac{1}{N}\sum_{i=1}^{N}\ |y_i - \hat{y}|$$ [34]

- **Mean Squared Error (MSE)**

The average of the squared difference between the original values from the test data set and values predicted by the model is represented by this. It is used for measuring the variance of the residuals. [5]

$$M.S.E\ = \frac{1}{N}\sum_{i=1}^{N}\ (y_t - \hat{y})^2$$ [34]

- **Root Mean Squared Error (RMSE)**

It is the result of the square root operator on Mean Squared Error. The standard deviation of residuals is measured using this. [5]

$$R.M.S.E\ = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\ (y_t - \hat{y})^2}$$ [34]

- **R2_score**

It is the or R-squared or coefficient of determination and is utilized to represent the proportion of the variance in dependent variables which is expressed by the regression model in other words how accurately the predicted values fit compared to the values of test. R-square is a scale-free score and so is not dependent on the size of the values. The R squared values from 0 to 1 and can be interpreted as percentages. [5]

$$r^2\ = \frac{\sum\ (y_t-\hat{y})^2}{\sum\ (y_i-\hat{y})^2}$$ [34]

In the above equations:

= mean value of y from the data set. [34]

= predicted values of y by the model. [34]

The residual is the difference between the observed mean value of y and the predicted value of y from the regression line. [5]

Chapter 3
Machine Learning Algorithms in Rice Yield Prediction based on Fertilizer, Land and Soil types

In this chapter, rice yield prediction methodology is used to predict the yield of Aus, Aman, and Boro rice using various agricultural parameters such as soil moisture, soil type, land type, and fertilizer. Moreover, we have worked using various machine learning algorithms to predict the rice yield of these three different rice varieties and in the next subchapters, we will be explaining in detail the working procedure like how we have trained and evaluated our model using machine learning algorithms as well as will be comparing the results.

3.1 Introduction

One of the most important sectors of Bangladesh is agriculture and it has a profound impact on the economy of Bangladesh. [35] There is much research available that has shown that the yield of crops mostly depends on the usage of fertilizers, different soil types, and soil moisture is also a dominant factor in maximizing the crop yield and it ensures efficient harvesting. [36] Basically, crop responses on these factors mostly. It is also shown in the various research paper that depending on the rate of fertilizer, soil type, the yield of crop varies from one place to another and it cannot be the same because the climate and weather conditions are not the same for all the places.[36] Moreover, yield rate also depends on the weather factors such as temperature, rainfall, humidity, wind speed, etc. but fertilizer, soil moisture, and land type those factors has the most impact on the yield of crops and these factors also changes from one location to another thus we can see that yield of rice like Aman or Boro is produced more in one location or district but in another district, we might have a low yield rate of that same crop. Hence, we get to know what type of land is responsible for the maximum crop yield and that can be determined from the structure of the soil. In this chapter, we are going to explore the yield of Aus, Aman, and Boro based on various fertilizers, soil types, and moisture using machine learning algorithms.

3.2 Related Work

There are plenty of works in the field of agriculture using machine learning to maximize the crop yield as this is the most vital sector of the economy for any country. So, we did some background research on the basis of our topic and found some papers about the yield of crops which are,

There is a journal similar to our research where the author highlights by his research that each crop responds almost in the same way based on the climate factors where the temperature was the most important factor that impacted the crop yield. Additionally, he mentioned that CO2 is also a prominent factor to predict how the crop is responding or the ways of increasing the crop response (D.L. Ehret, et al., 2011). [37]

We have found another research paper where the author developed a management system for agriculture and he mentioned in his paper that proper and accurate techniques should be developed for better estimation in order to identify how accurately and effectively machine learning models can predict the rice yield in different region or district depending on the weather conditions and this paper also talks about the evaluation of models using agricultural parameters as well as compares various artificial neural network models to find out which technique is more accurate and efficient in yield prediction(B. Jiet. Al., 2007). [38]

3.3 Overview of Dataset

Dataset is a fundamental thing to develop an efficient prediction system of rice yield. Any machine learning-based system gathers knowledge from the dataset which further allows predicting the yield in the near future. In our project, we have taken the dataset from Kaggle named ‘Agricultural dataset' and it was publicly available to use. [39] From the top crops of Bangladesh, we are working with Aus rice, Boro rice, and Aman rice thus our research mainly focuses on the further exploration of these three major crops of Bangladesh. All the necessary data for predicting rice yield were compiled into tabulated form in this dataset. We normalized the values of all the parameters by dividing each set of parameters by the largest value of that set. This gives us values ranging from 0 to 1. As we generally use .csv files of the dataset in the machine learning model implementation thus we converted Excel format into a comma-separated value (CSV) format to use it in the model. [40]

Though various environmental parameters such as rainfall, minimum and maximum temperature, humidity, etc. are present in the dataset these parameters have less impact on the yield of Aus, Aman, and Boro rice which will be described in the subsequent chapters. Besides, several fertilizers were considered in this dataset which is responsible for the maximum yield and inundation land types were categorized in the form of highland, medium lowland, medium highland, lowland, and very lowland. Additionally, soil type, structure, and soil moisture were also included in the dataset which has a great impact on the cultivation.

In figure 3.1(a), we can see the average production of Aus rice between the years 2008 to 2017 in seven different districts where Mymensingh has the highest Aus rice production and Kishoreganj also showed a preferable amount of rice production, as well as Narsingdi, has the lowest Aus rice production.

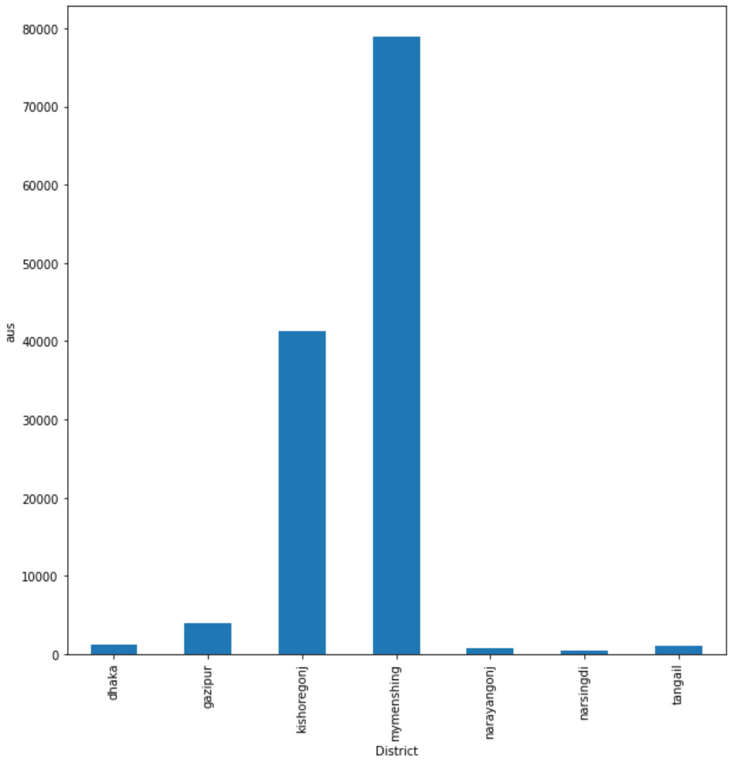


Figure 3.1(a): District wise Aus rice production

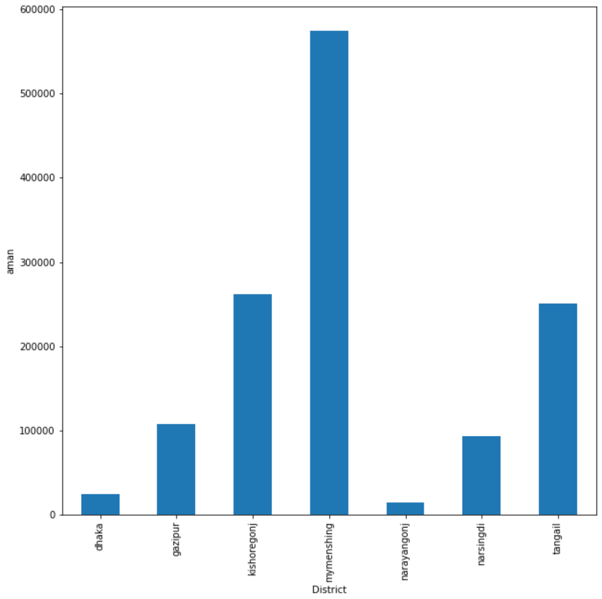


Figure 3.1(b): District wise Aman rice production

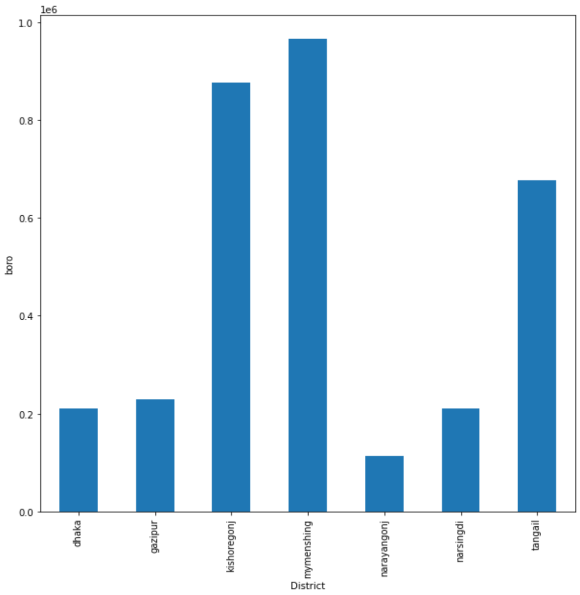


Figure 3.1(c): District wise Boro rice production

Figure 3.1: District wise Aman, Aus, and Boro Production

In figure 3.1(b), the graph indicates that the average production of Aman rice between the years 2008 to 2017 in seven different districts where Mymensingh still has the highest Aman rice production and Kishoreganj also produced a high amount of Aman rice.

In Addition, figure 3.1(c) indicates the crop yield of Boro rice between the years 2008 to 2017 in seven different districts where it is seen that the yield in Dhaka and Tangail has been increased than Aus and Mymensingh is still the leading district in Boro rice production than any other district and the Boro yield production rate in Kishoreganj has also increased compared to other crop yield production in this district.

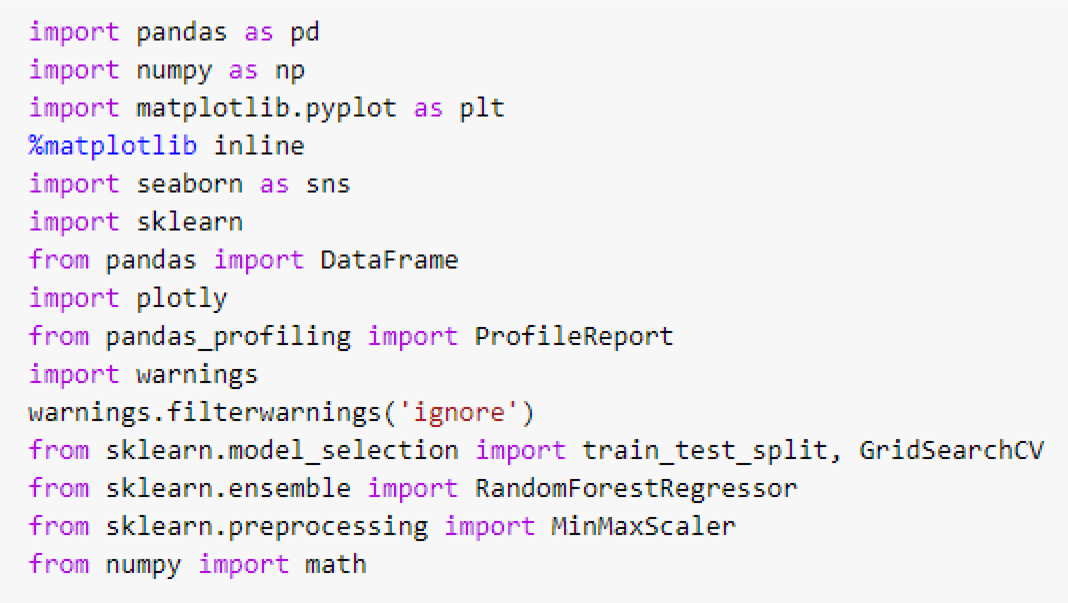
3.4 General Procedure

3.4.1 Data preprocessing

Data preprocessing is a very basic and significant step in the implementation of machine learning models. It basically means processing the data by dropping or manipulating the dataset before using it in the system to get better performance and this process has a great impact in enhancing the overall accuracy of the model. [41] The steps are described below,

Importing Library

The First stage is to import necessary library functions and modules for the model implementation. The Panda library in python was used to read and display the dataset by which we can understand the characteristics of the dataset. [41]

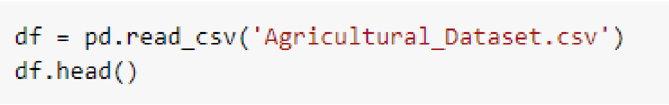


```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import sklearn
from pandas import DataFrame
import plotly
from pandas_profiling import ProfileReport
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import MinMaxScaler
from numpy import math
```

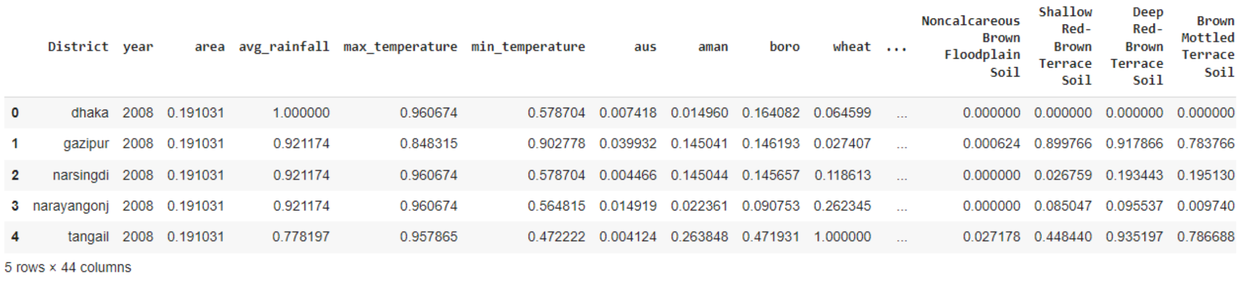
Here, the 'import pandas as pd' library function is used to view the data in the data frame. NumPy is used to create a multidimensional array. Moreover, the Matplotlib function is used for plotting the graphs in python and the MinMaxScaler function is used for feature scaling which helps to enhance the accuracy of the implemented model and also reduces the error rate. Moreover, the Train test split function is imported to split the dataset into the training and testing parts as well as the Random forest regressor function is also imported from the Scikit learn library to develop the model using the Random Forest algorithm. [41]

Importing dataset

We have used the command 'pd.read_csv()' to read the data into a data frame.



```
df = pd.read_csv('Agricultural_Dataset.csv')
df.head()
```

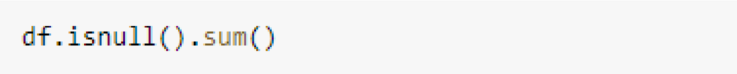


	District	year	area	avg_rainfall	max_temperature	min_temperature	aus	aman	boro	wheat	...	Noncalcareous Brown Floodplain Soil	Shallow Red-Brown Terrace Soil	Deep Red-Brown Terrace Soil	Brown Mottled Terrace Soil
0	dhaka	2008	0.191031	1.000000	0.960674	0.578704	0.007418	0.014960	0.164082	0.064599	...	0.000000	0.000000	0.000000	0.000000
1	gazipur	2008	0.191031	0.921174	0.848315	0.902778	0.039932	0.145041	0.146193	0.027407	...	0.000624	0.899766	0.917866	0.783766
2	narsingdi	2008	0.191031	0.921174	0.960674	0.578704	0.004466	0.145044	0.145657	0.118613	...	0.000000	0.026759	0.193443	0.195130
3	narayangonj	2008	0.191031	0.921174	0.960674	0.564815	0.014919	0.022361	0.090753	0.262345	...	0.000000	0.085047	0.095537	0.009740
4	tangail	2008	0.191031	0.778197	0.957865	0.472222	0.004124	0.263848	0.471931	1.000000	...	0.027178	0.448440	0.935197	0.786688

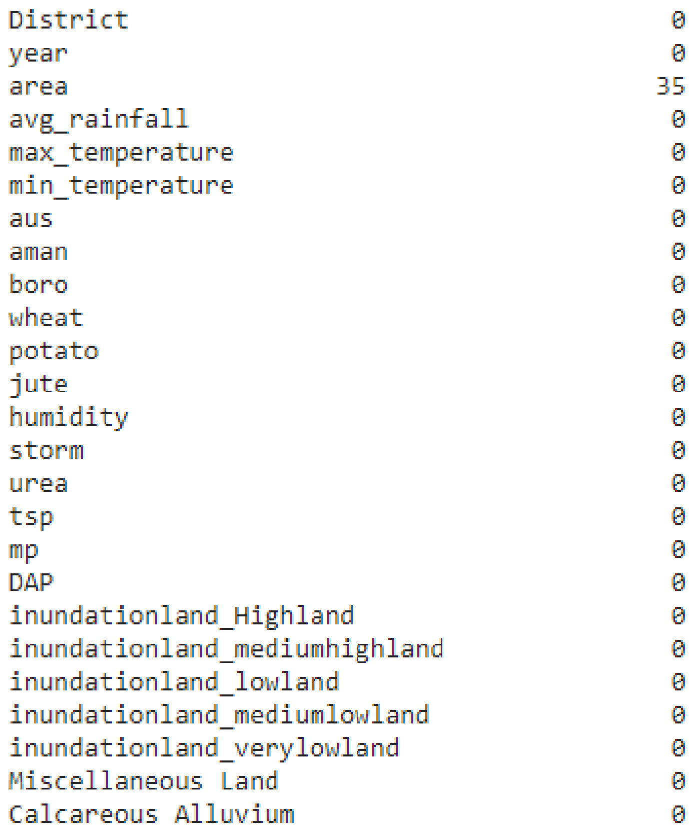
5 rows × 44 columns

Handling of Missing value

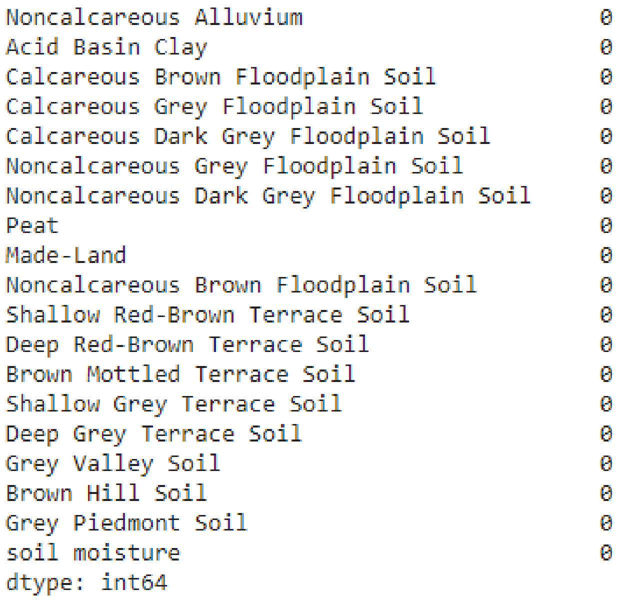
To find out the missing values in Pandas, we use the function 'isnull()' which helps us to know whether there are any null values in the dataset or not. [41]



```
df.isnull().sum()
```



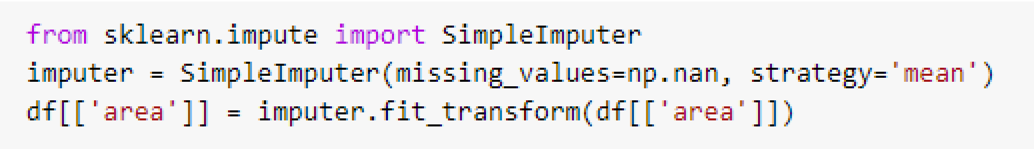
```
District                                     0
year                                         0
area                                        35
avg_rainfall                                 0
max_temperature                              0
min_temperature                              0
aus                                          0
aman                                         0
boro                                         0
wheat                                        0
potato                                       0
jute                                         0
humidity                                     0
storm                                        0
urea                                         0
tsp                                          0
mp                                           0
DAP                                          0
inundationland_Highland                      0
inundationland_mediumhighland                0
inundationland_lowland                       0
inundationland_mediumlowland                 0
inundationland_verylowland                   0
Miscellaneous Land                           0
Calcareous Alluvium                          0
```



```
Noncalcareous Alluvium                      0
Acid Basin Clay                             0
Calcareous Brown Floodplain Soil            0
Calcareous Grey Floodplain Soil             0
Calcareous Dark Grey Floodplain Soil        0
Noncalcareous Grey Floodplain Soil          0
Noncalcareous Dark Grey Floodplain Soil     0
Peat                                        0
Made-Land                                   0
Noncalcareous Brown Floodplain Soil         0
Shallow Red-Brown Terrace Soil              0
Deep Red-Brown Terrace Soil                 0
Brown Mottled Terrace Soil                  0
Shallow Grey Terrace Soil                   0
Deep Grey Terrace Soil                      0
Grey Valley Soil                            0
Brown Hill Soil                             0
Grey Piedmont Soil                          0
soil moisture                               0
dtype: int64
```

As it is shown in the output image that except for the parameter ‘area’, we do not have any other missing or null values.

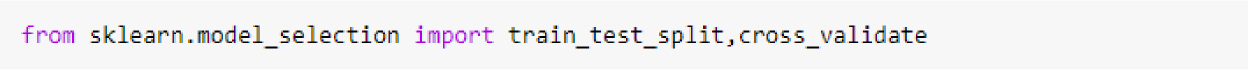
Now, to replace or impute the missing value with a constant value, we use the SimpleImputer class by which missing values will be imputed easily.



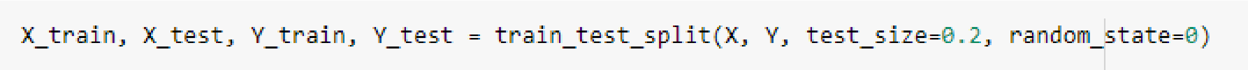
```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
df[['area']] = imputer.fit_transform(df[['area']])
```

Training & Test Split

The train test split is used for dividing the dataset into two subsets where there will be a training part and another part will be the testing part. We have used a ratio of 80 and 20 where 80% of the data is assigned for the training part and 20% data for the testing part which we did by mentioning the test size as ‘0.2’.



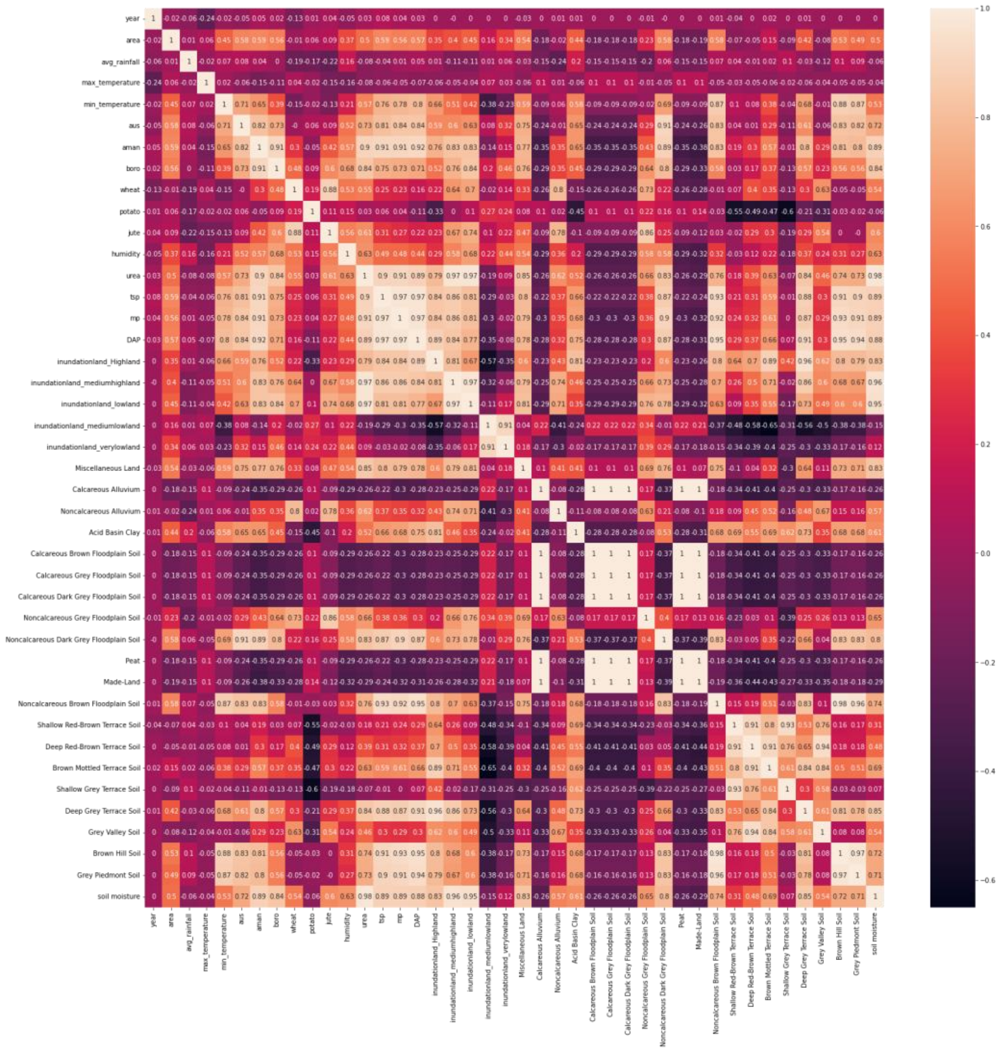
```
from sklearn.model_selection import train_test_split,cross_validate
```



```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=0)
```

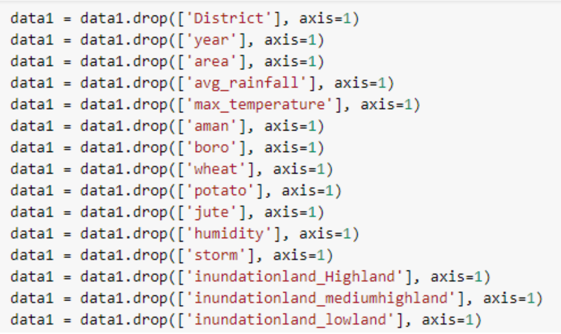
3.4.2 Pearson's correlation heatmap for AUS

A correlation heatmap is basically the graphical representation of the correlation matrix between various variables. [40] In our case, we have used the heatmap to show the correlation between Aus and other parameters such as maximum and minimum temperature, rainfall, various types of land like medium highland, medium lowland, very lowland, etc and various types of soil as well as fertilizer. With the help of this heatmap, we can determine which variables or factors have the most influence on the Aus, Aman, and Boro yield.

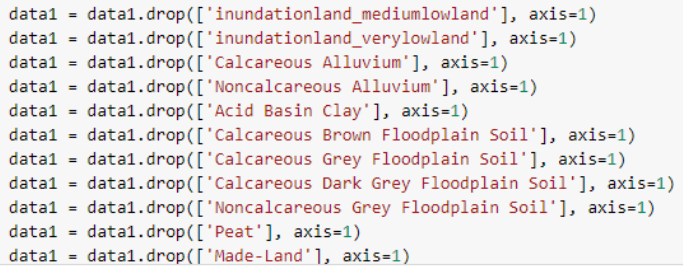


For a strong correlation, the value has to be close to -1 for the negative correlation and close to +1 for the positive correlation. We have selected the threshold value to be 0.7 for the positive correlation and -0.7 for the negative correlation and therefore we have dropped all the variables which are below that threshold value.

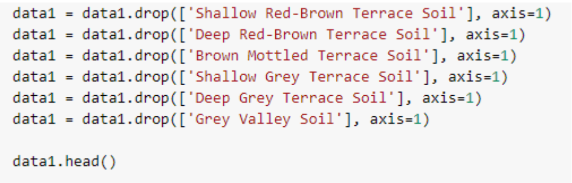
In the case of Aus, we have checked the variables which have a positive or negative correlation with Aus and dropped the other variables which do not have any correlation and we will follow the same procedure for Aman and Boro respectively.



```
data1 = data1.drop(['District'], axis=1)
data1 = data1.drop(['year'], axis=1)
data1 = data1.drop(['area'], axis=1)
data1 = data1.drop(['avg_rainfall'], axis=1)
data1 = data1.drop(['max_temperature'], axis=1)
data1 = data1.drop(['aman'], axis=1)
data1 = data1.drop(['boro'], axis=1)
data1 = data1.drop(['wheat'], axis=1)
data1 = data1.drop(['potato'], axis=1)
data1 = data1.drop(['jute'], axis=1)
data1 = data1.drop(['humidity'], axis=1)
data1 = data1.drop(['storm'], axis=1)
data1 = data1.drop(['inundationland_Highland'], axis=1)
data1 = data1.drop(['inundationland_mediumhighland'], axis=1)
data1 = data1.drop(['inundationland_lowland'], axis=1)
```



```
data1 = data1.drop(['inundationland_mediumlowland'], axis=1)
data1 = data1.drop(['inundationland_verylowland'], axis=1)
data1 = data1.drop(['Calcareous Alluvium'], axis=1)
data1 = data1.drop(['Noncalcareous Alluvium'], axis=1)
data1 = data1.drop(['Acid Basin Clay'], axis=1)
data1 = data1.drop(['Calcareous Brown Floodplain Soil'], axis=1)
data1 = data1.drop(['Calcareous Grey Floodplain Soil'], axis=1)
data1 = data1.drop(['Calcareous Dark Grey Floodplain Soil'], axis=1)
data1 = data1.drop(['Noncalcareous Grey Floodplain Soil'], axis=1)
data1 = data1.drop(['Peat'], axis=1)
data1 = data1.drop(['Made-Land'], axis=1)
```



```
data1 = data1.drop(['Shallow Red-Brown Terrace Soil'], axis=1)
data1 = data1.drop(['Deep Red-Brown Terrace Soil'], axis=1)
data1 = data1.drop(['Brown Mottled Terrace Soil'], axis=1)
data1 = data1.drop(['Shallow Grey Terrace Soil'], axis=1)
data1 = data1.drop(['Deep Grey Terrace Soil'], axis=1)
data1 = data1.drop(['Grey Valley Soil'], axis=1)

data1.head()
```

Thus, the correlation heatmap looks like this:

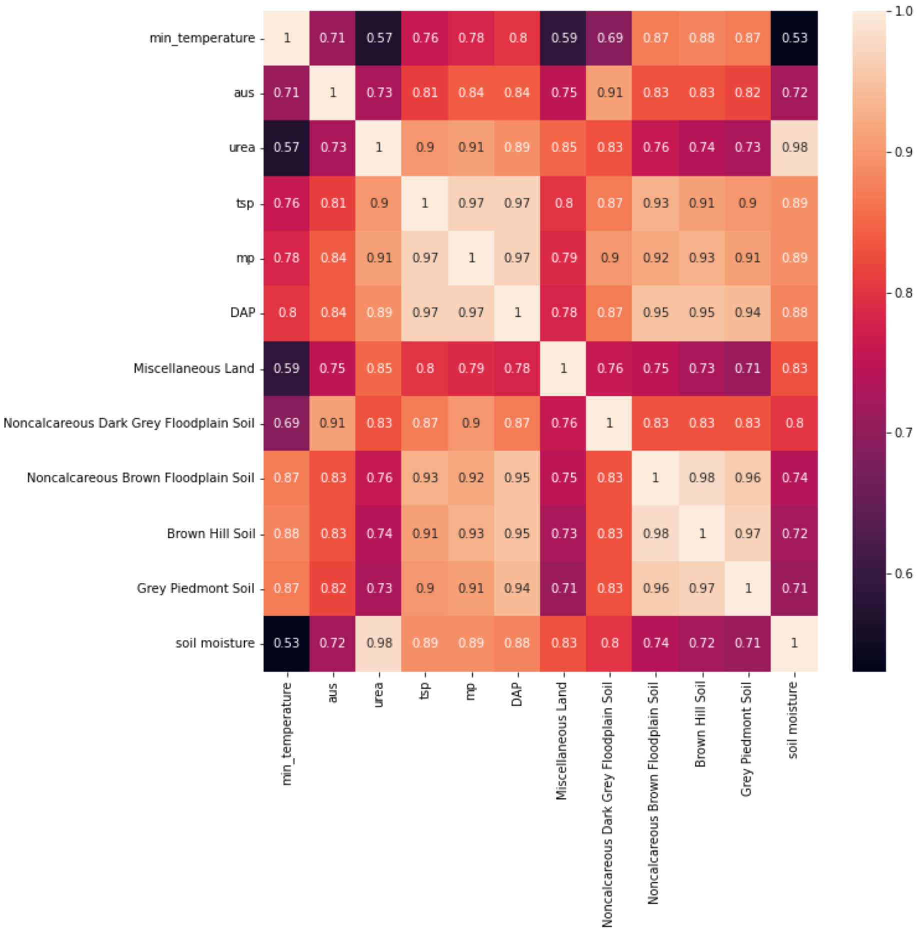


Figure 3.2(a): correlation heatmap for AUS

3.4.3 Pearson’s correlation heatmap for AMAN

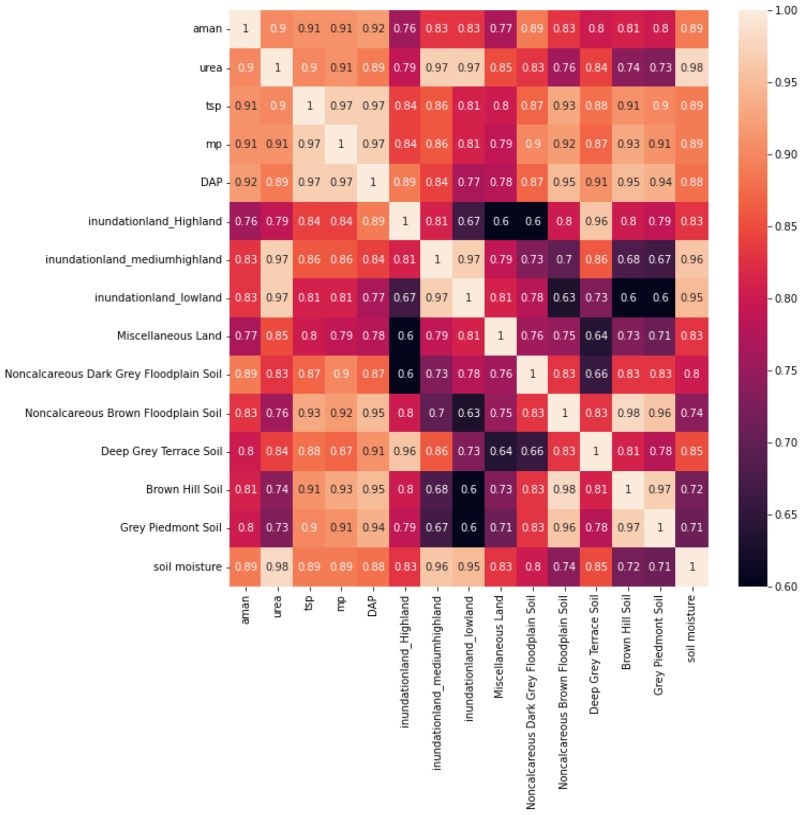


Figure 3.2(b): correlation heatmap for Aman

3.4.4 Pearson’s correlation heatmap for BORO

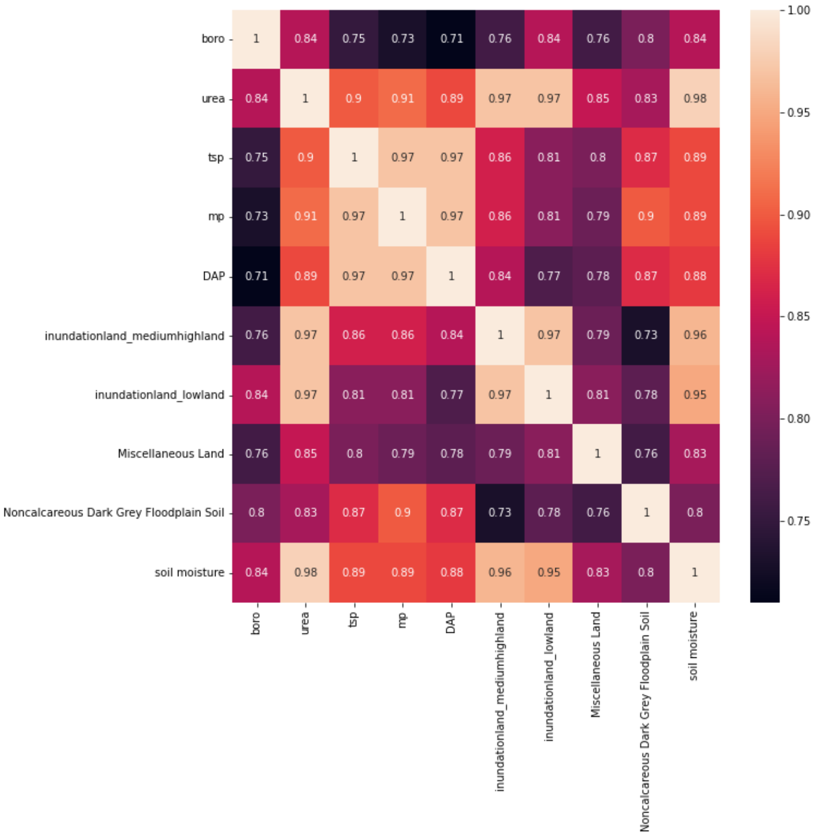


Figure 3.2(c): correlation heatmap for Boro

Figure 3.2: correlation heatmap

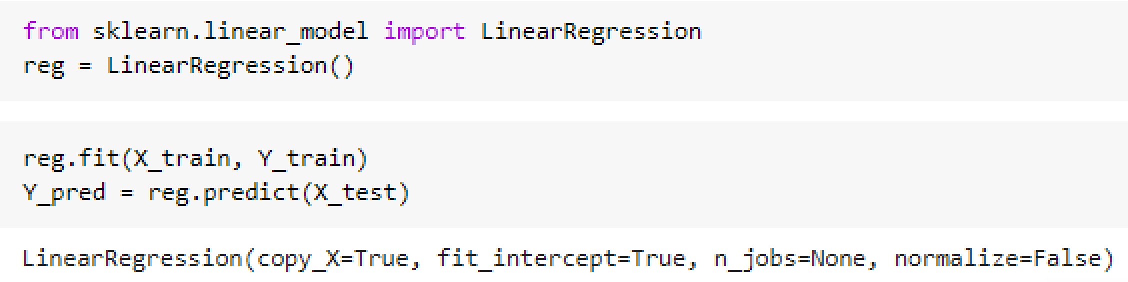
3.5 Model Implementation

As we will predict and implement the model of three types of major crops in Bangladesh, to predict the yield of each crop with machine learning algorithms, we have to work with a single crop at a time, and in order to do that, we will drop the other crops for example, when we implement the model for Aus yield, we will drop Aman and Boro from the dataset and we will implement four separate machine learning model or algorithm to predict the yield. Similarly, we will implement the models for Aman and Boro separately and without the crop itself and correlated parameters, other variables will be dropped from the dataset.

With the aim of predicting the yield of the crop, after preprocessing the data we need to fit that dataset into a model to see the result of our prediction as well as how accurate our result is, and also, we need to evaluate the loss of our predicted result. For that purpose, we have used four different algorithms and implemented the models using Linear regression, Random forest regressor, K-nearest neighbor regression, and Gradient boosting algorithm to predict the result and with the help of various performance metrics, we can also assess the performance of each machine learning models.

1. Linear Regression

We have first used the linear regression algorithm which is a very basic and easy machine learning technique to apply for prediction though this algorithm is not efficient all the time and the performance of the model may not be as high as expected. For our agricultural dataset, we have used multiple linear regression as we were not working with binary-type variables. A regression model is basically implemented by showing the relationship between dependent variable x and independent variable y. Additionally, a non-linear relationship between two variables is fitted in the regression model for evaluating the accuracy of this algorithm. [14] To fit this regression algorithm in our dataset, we have used a command reg.fit() and thus implemented the model. Then after fitting the model, we need to predict the result in the testing dataset which is shown below,



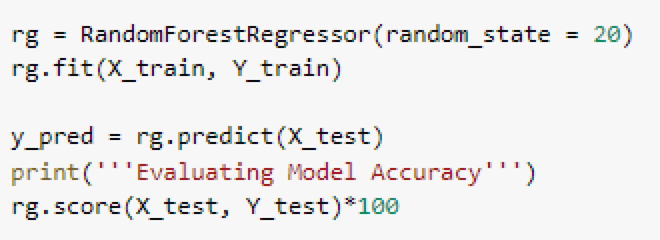
```
from sklearn.linear_model import LinearRegression
reg = LinearRegression()
```

```
reg.fit(X_train, Y_train)
Y_pred = reg.predict(X_test)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

2. Random Forest

Random forest algorithm is a combination of decision trees and it is basically a classification model. Random forest is very efficient in order to get a very accurate prediction and the performance of the model implemented using this algorithm is generally high. In order to get a better prediction, this algorithm is used as it collects the training data from all nodes and by separating the nodes which are weaker it helps to get better results.[14] The accuracy of the model is basically predicted from the testing data and the result of model performance or prediction is dependent on the size of training and testing data in many cases as it can make differences in the results. This algorithm is used to solve classification problems as well as regression problems. [14]

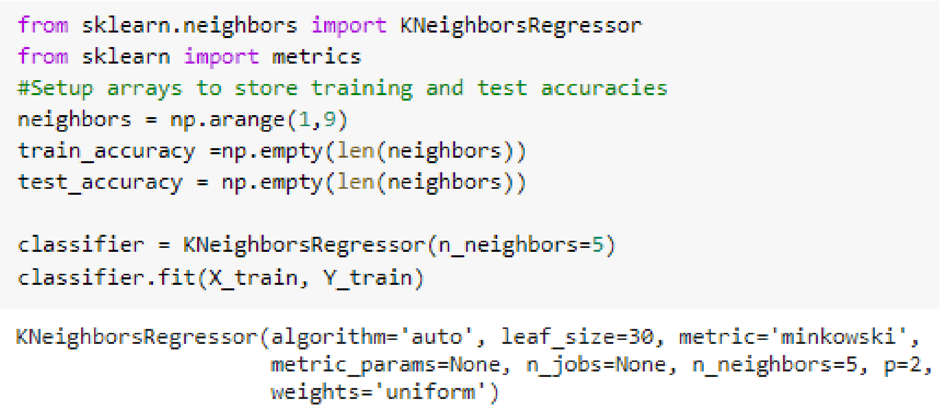


```
rg = RandomForestRegressor(random_state = 20)
rg.fit(X_train, Y_train)

y_pred = rg.predict(X_test)
print('''Evaluating Model Accuracy''')
rg.score(X_test, Y_test)*100
```

3. K-nearest neighbors

K nearest neighbor is one of the most efficient algorithms in machine learning which helps to get better predictions and results, as well as the performance of the model is very high. In our research, we have implemented this KNN model for Aus, Aman, and Boro separately to predict the result and the accuracy of this model is mostly high compared to any other algorithms. To fit a dataset into the KNN model, we need to select a variable which neighbors(k) and for the nearest neighbor of the KNN model, the performance of the implemented model will be higher. To find out which neighbor is closer to the K nearest neighbor, we have changed the value of 'k' multiple times and selected the value for which we get the higher accuracy. For example, in the case of Aus, the value of our neighbor was 2 but for Aman and Boro, we got the higher accuracy when the value of 'k' was 2. Thus, the KNN algorithm works efficiently to get better predictions.



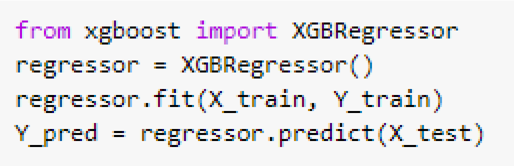
```
from sklearn.neighbors import KNeighborsRegressor
from sklearn import metrics
#Setup arrays to store training and test accuracies
neighbors = np.arange(1,9)
train_accuracy =np.empty(len(neighbors))
test_accuracy = np.empty(len(neighbors))

classifier = KNeighborsRegressor(n_neighbors=5)
classifier.fit(X_train, Y_train)
```

```
KNeighborsRegressor(algorithm='auto', leaf_size=30, metric='minkowski',
                    metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                    weights='uniform')
```

4. Extreme Gradient Boosting

Gradient boosting algorithm is an influential machine learning technique that is used for regression analysis and also for solving various classification problems. In our research, as we are working with regression problems, we have used the regression form of this algorithm. Moreover, this system is a boosting method so this algorithm basically made the predictions that are not independent but sequential. [8] This method observes the mistakes which were previously made by the predictors. [8] and therefore the prediction is more accurate and to determine the accuracy of the model, we usually find the loss function which is the mean squared error, and if the error of the model is lower, then the model is capable of predicting the results better.



```
from xgboost import XGBRegressor
regressor = XGBRegressor()
regressor.fit(X_train, Y_train)
Y_pred = regressor.predict(X_test)
```

3.6 Result & Discussion

In this part of this chapter, we will evaluate the results of different algorithms in the prediction of Aus, Aman, and Boro yield. After implementing different machine learning algorithms, we have performed the accuracy test, error, and scatter plotting of the actual value with respect to our predicted value which determines the performance of the models.

3.6.1 Accuracy of the models for Aus

Accuracy is an important metric for any algorithm as it determines the performance of the model. Getting better accuracy means this model is capable of giving better predictions.

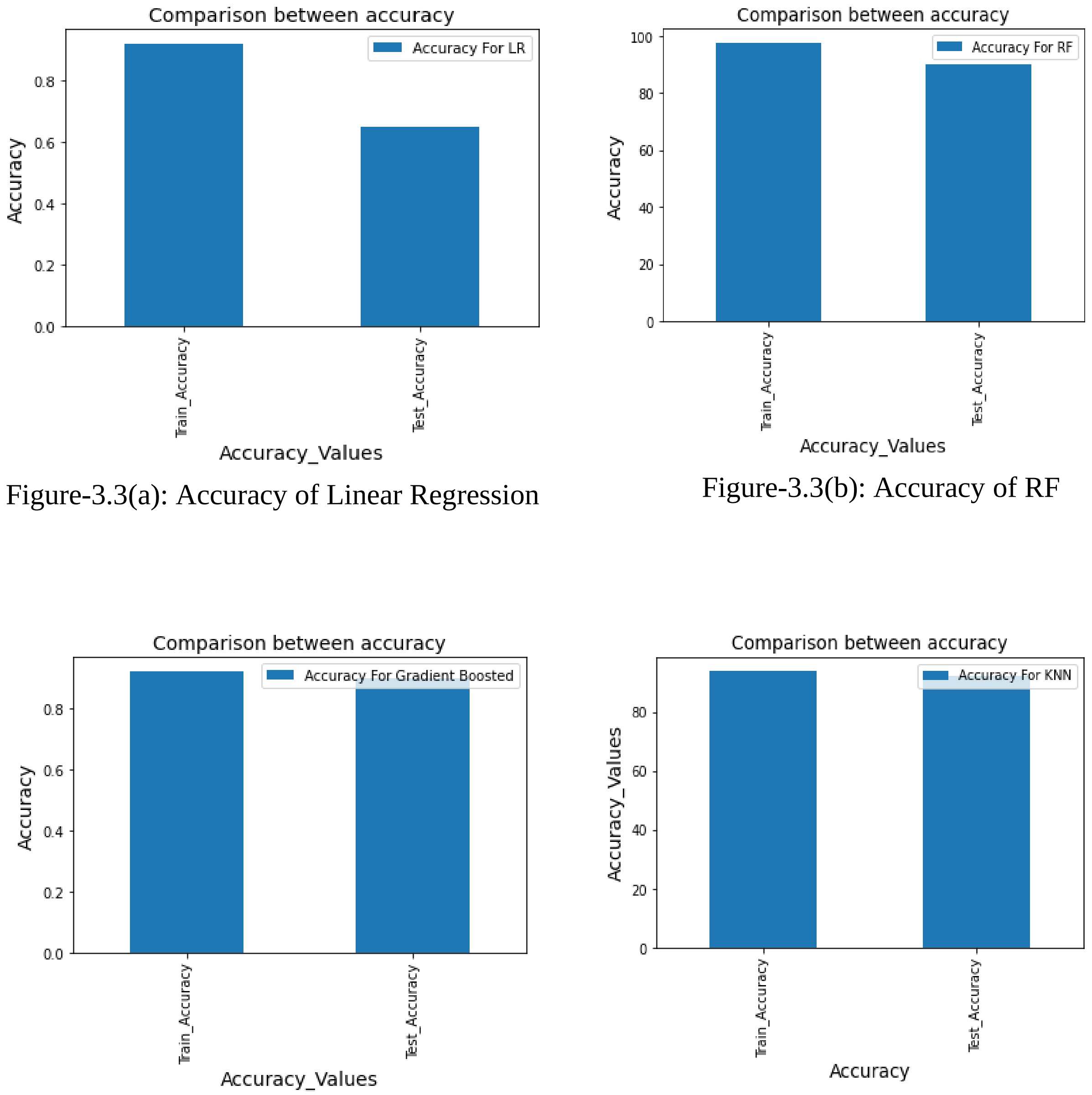


Figure-3.3(a): Accuracy of Linear Regression

Figure-3.3(b): Accuracy of RF

Figure-3.3(c): Accuracy of GB

Figure-3.3(d): Accuracy of KNN

Figure-3.3: Accuracy of different algorithms

In figure-3.3(a), the accuracy of the linear regression model for Aus is displayed above where the training accuracy is 92% and this model has a testing accuracy of 65%. As we evaluate the overall performance of the model from the testing accuracy so in that case, linear regression has less accuracy compared to other algorithms.

In figure-3.3(b), the accuracy of the Random Forest model for Aus is visible and this model has a training accuracy of 98% as well as the testing accuracy is 90%.

In figure-3.3(c), from the Gradient boosting model, we get the training accuracy of 92% and this model has a testing accuracy of 90%.
In figure-3.3(d), the accuracy of the KNN model for Aus is shown where the training accuracy is 94% and this model has a testing accuracy of 92% which is high among all algorithms.

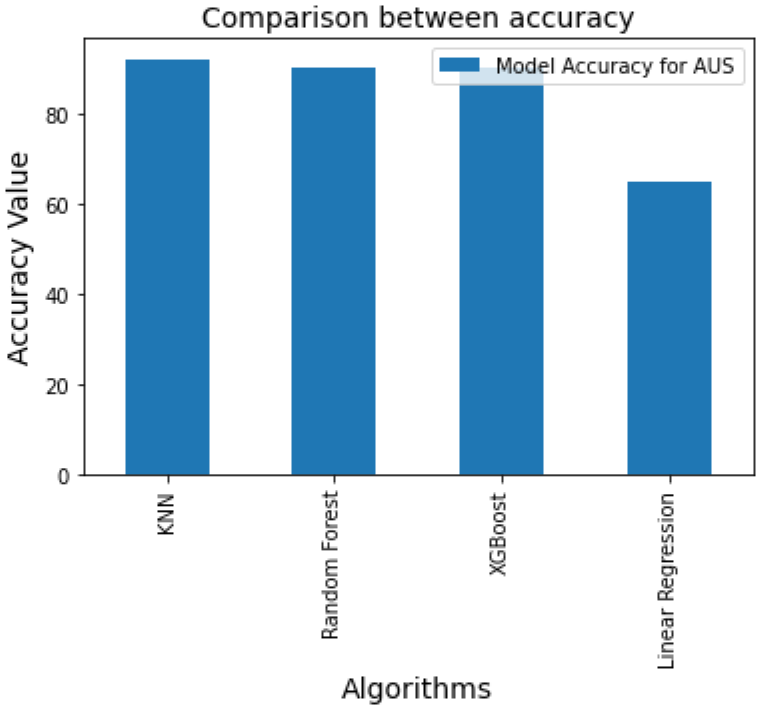


Figure-3.4: Comparison between the accuracy of different algorithms for Aus

We have plotted the Accuracy vs Algorithm graph for Aus in figure-3.4 to differentiate the model accuracy of different algorithms. From the above picture, we can see that the accuracy of the KNN model is higher than other algorithms which is 92%.

3.6.2 Errors of the models for Aus

Model's performance is also dependent on the values of error or loss of a model. The higher the error of a model, the lower will be the model performance, as well as the ability to predict results, will be poor.

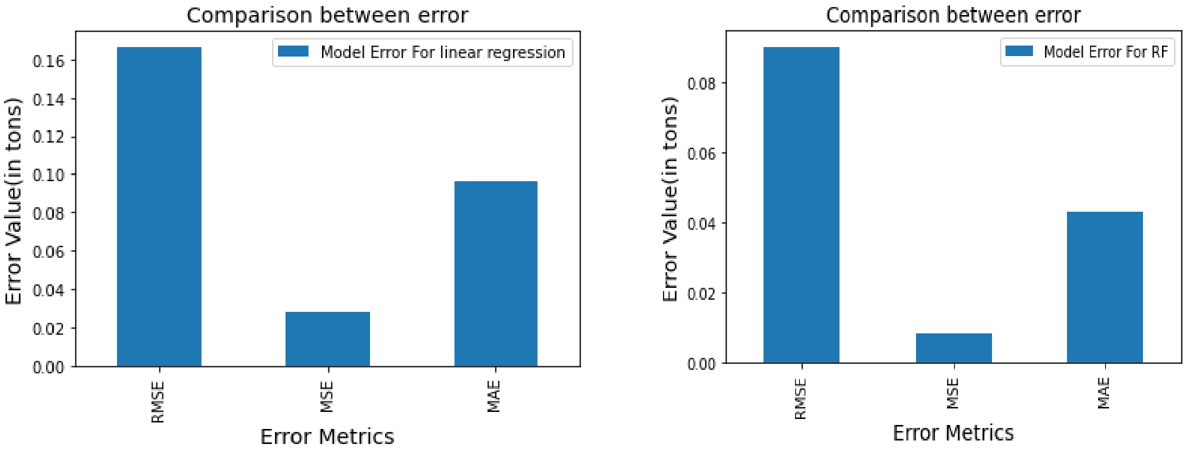


Figure-3.5(a): Error of Linear regression

Figure-3.5(b): Error of RF

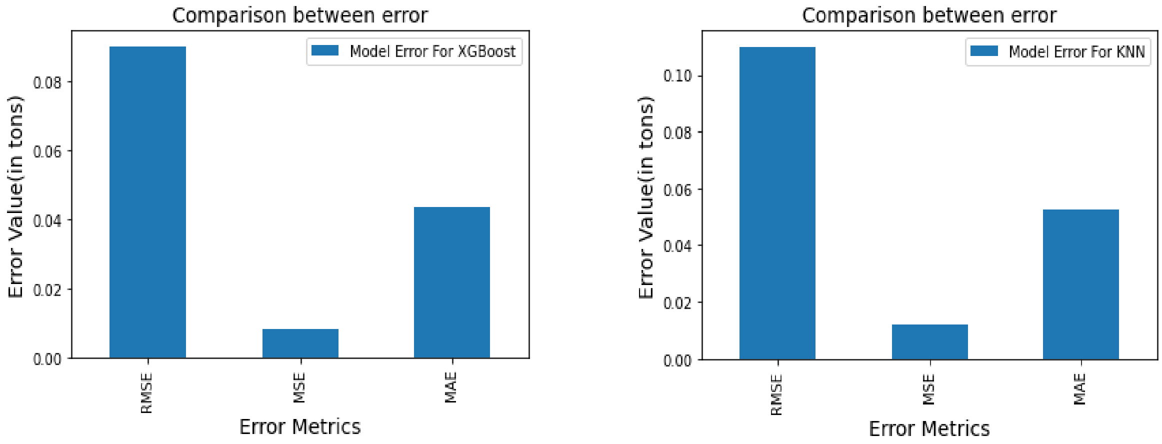


Figure-3.5(c): Error of XGB

Figure-3.5(d): Error of KNN

Figure-3.5: Error (in tons) of algorithms for Aus

In figure-3.5(a), errors of the Linear regression model are shown where the value of RMSE is almost double than MAE and MSE is lower than other error factors in this model.
In figure-3.5(b), the RMSE value of the Random forest model is much bigger and we can see that the MSE of this model is lower than the linear regression model.
In figure-3.5(c), the RMSE value is higher but similar to the other models and MSE, MAE is almost similar to the random forest model.
In figure-3.5(d), we can see that the RMSE value of the KNN model is much bigger compared to the mean absolute error or mean squared error of the model.

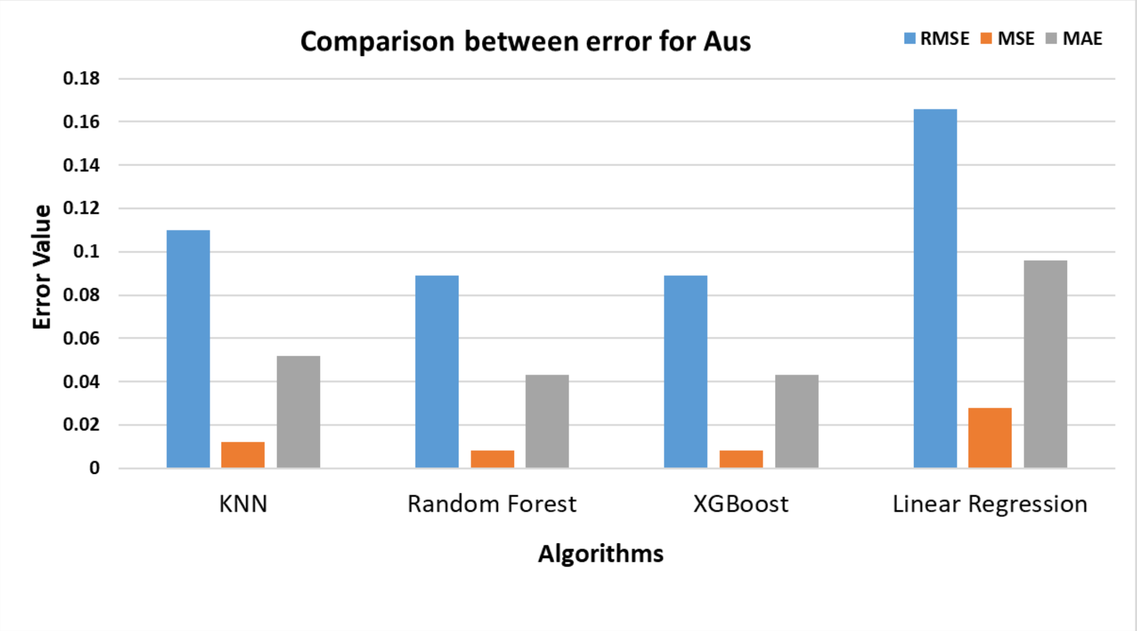


Figure-3.6: Comparison between Errors (in tons) of algorithms

To see the changes of error in the graphs, we have plotted the root mean square error, mean absolute percentage error, and mean absolute error in the graph for all the algorithms. We also found the value of mean squared error but it is too big to plot in the graph. From the graph, we can observe that the error values such as RMSE, MSE for KNN are much lower than the others. and MAE is not visible because it’s too low compared to other’s error values. Moreover, MAE of linear regression and k nearest neighbor model is higher than the random forest and XGB models.

3.6.3 Scatter plot for Aus

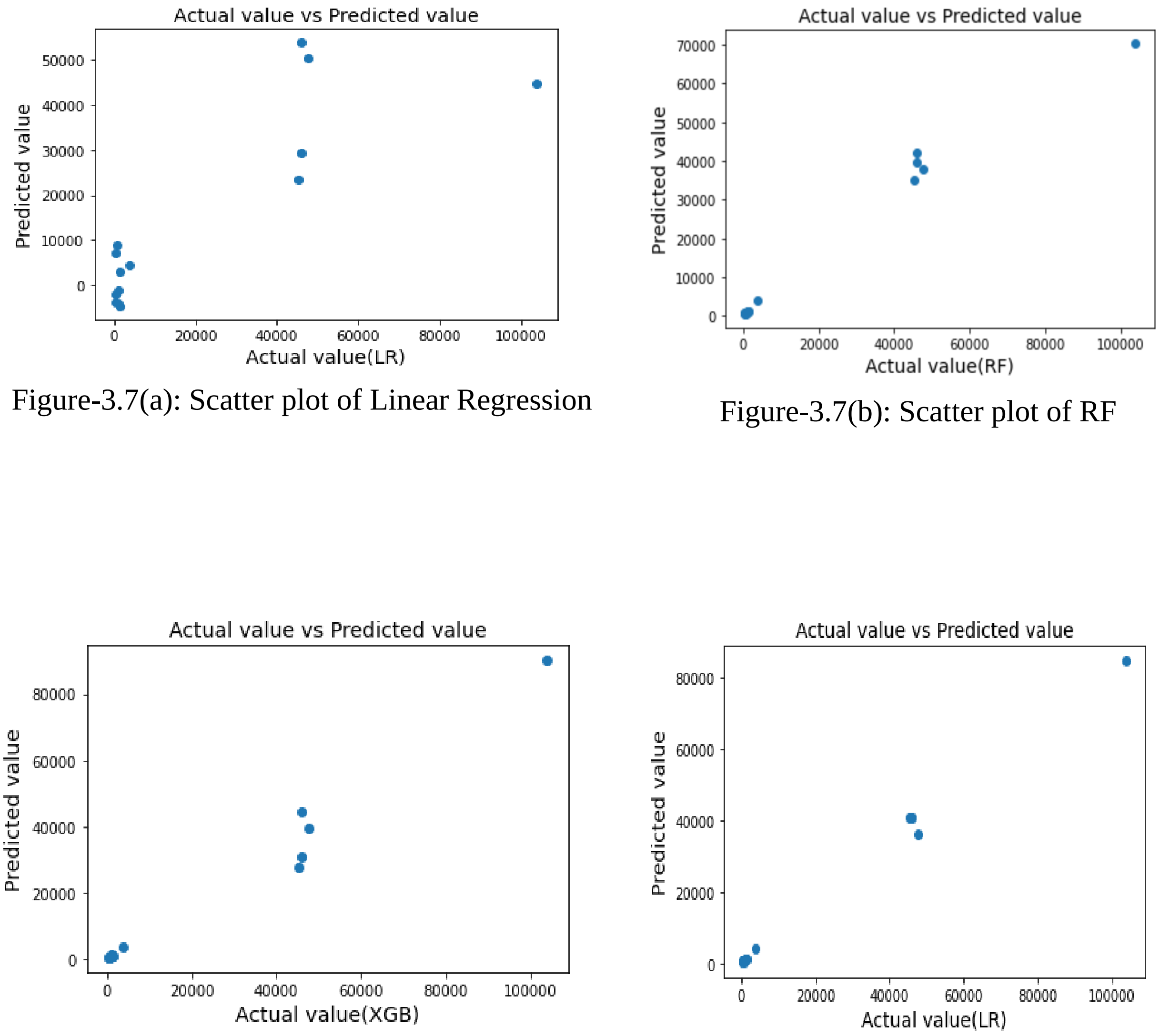


Figure-3.7(a): Scatter plot of Linear Regression

Figure-3.7(b): Scatter plot of RF

Figure-3.7(c): Scatter plot of XGB

Figure-3.7(d): Scatter plot of KNN

Figure-3.7: Scatter plot for Aus

In this part of our research, we have a very small dataset and for this reason, it's showing a very low number of variables in the scatter plot. Although the quantity of data entries was low, those variables like fertilizer, soil type, and soil moisture have very good correlation with Aus and these factors have impacted the yield of Aus, Aman, and Boro.

With the help of scattering plotting, we can see the actual value of our dataset against the value we have predicted by implementing the model. In this case, the actual values will be on the x-axis and the predicted values will be on the y-axis. Scatter plots are mainly used in the linear regression problems but as we have used the regression type of every algorithm so we have plotted the scatter form of all the algorithms to observe how much closer the values are to each other and also to the regression line.

For the above pictures, we can see that values are not much closer to each other but if we try to draw a regression line then the values are much closer to that line.

3.6.4 Accuracy of the models for Aman

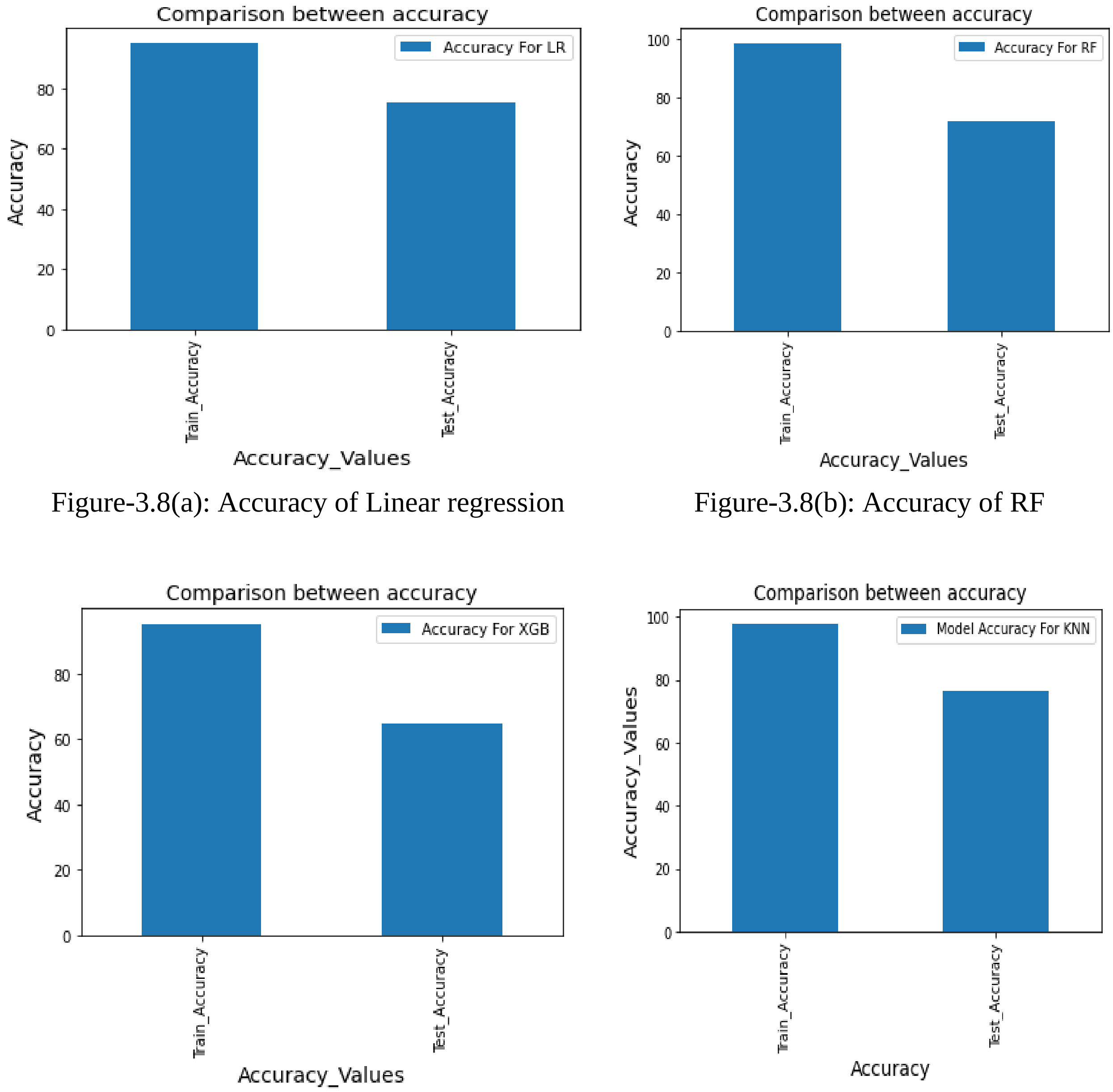


Figure-3.8(a): Accuracy of Linear regression

Figure-3.8(b): Accuracy of RF

Figure-3.8(c): Accuracy of GB

Figure-3.8(d): Accuracy of KNN

Figure-3.8: Accuracy plot for Aman

In figure-3.8(a), the accuracy of the linear regression model for Aman is shown where the training accuracy is 95% and this model has a testing accuracy of 75% which is high among all algorithms. In figure-3.8(b), from the Random Forest model, we get the training accuracy of 99% and this model has a testing accuracy of 73%.

In figure-3.8(c), the accuracy of the gradient boosting model for Aman is displayed above where the training accuracy is 95% and this model has a testing accuracy of 65%.

In figure-3.8(d), the accuracy of the KNN model for Aman is visible and this model has a training accuracy of 98% as well as the testing accuracy is 76%.

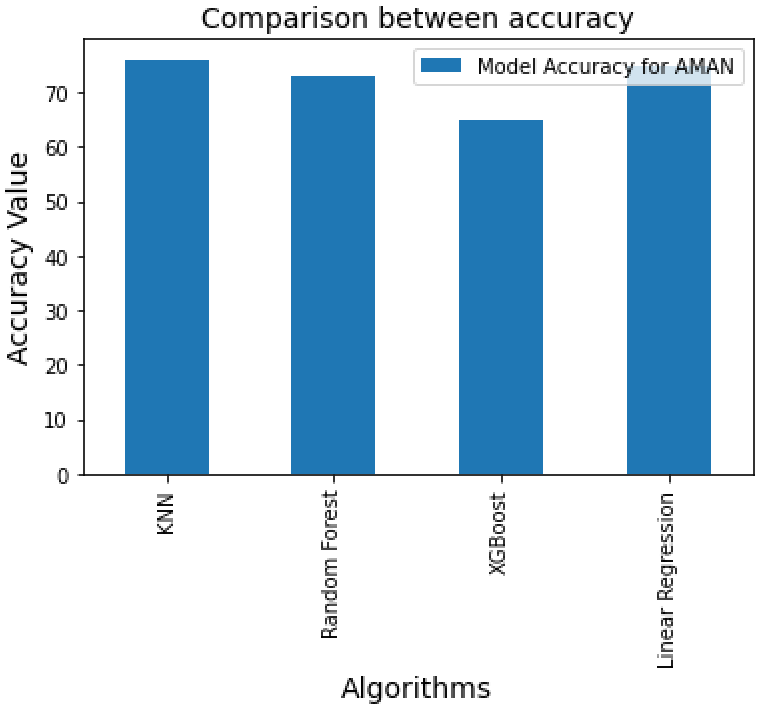


Figure-3.9: Comparison between Accuracy for Aman

We have plotted the Accuracy vs Algorithm graph for Aus in figure-3.4 to differentiate the model accuracy of different algorithms. From the above picture, we can see that the accuracy of the KNN model is higher than other algorithms which is 76%

3.6.5 Errors of the models for Aman

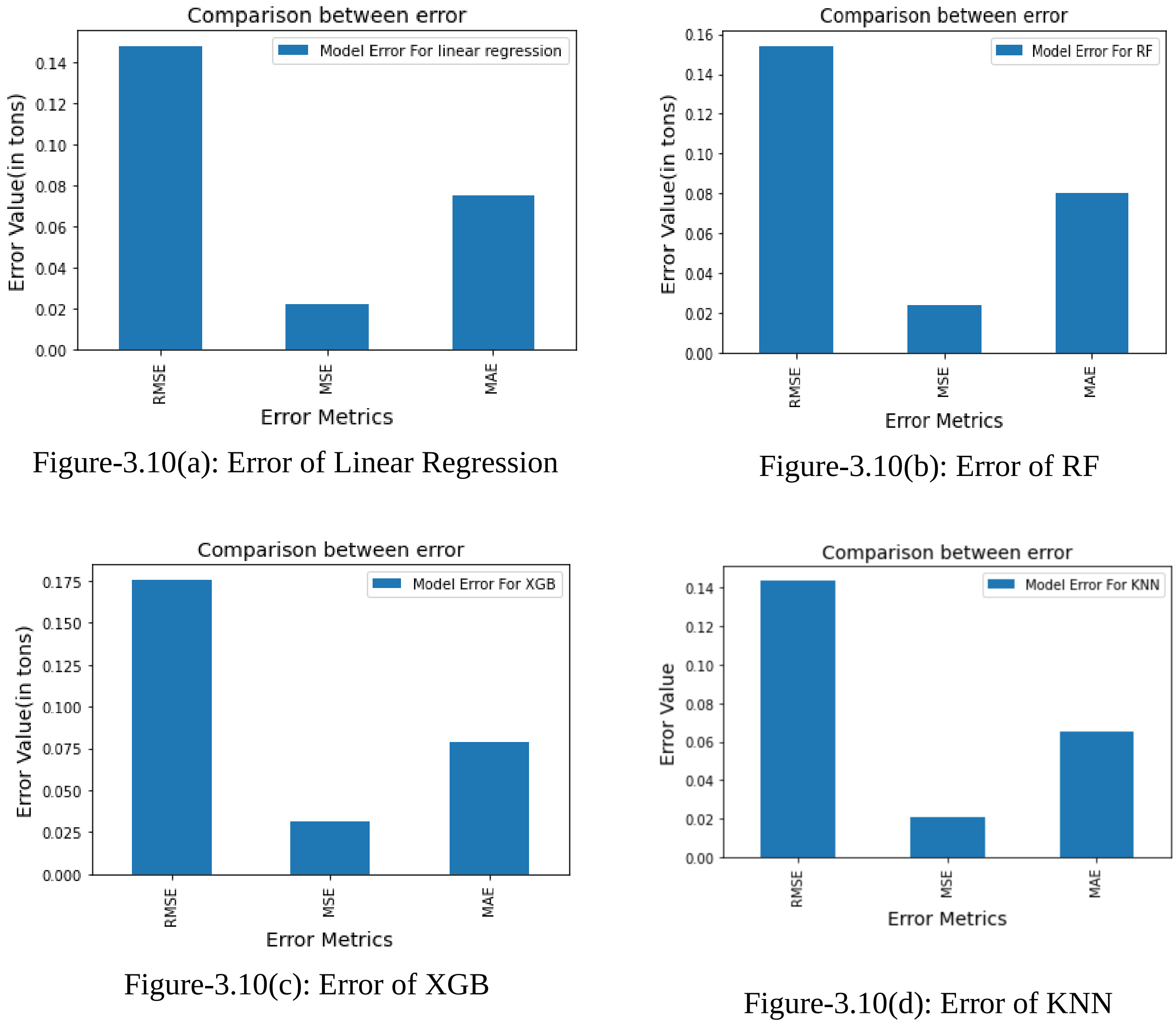


Figure-3.10(a): Error of Linear Regression

Figure-3.10(b): Error of RF

Figure-3.10(c): Error of XGB

Figure-3.10(d): Error of KNN

Figure-3.10: Error for Aman

In figure-3.10(a), we can see that the RMSE value of the linear regression model is much bigger compared to the mean absolute error or mean squared error of the model.

In figure-3.10(b) errors of the Random forest model are shown where the value of RMSE is almost double than MAE and MSE is lower than other errors in this model.

In figure-3.10(c), the gradient boosting model has the RMSE value is higher than other models and MSE is much lower than RMSE and MAE.

In figure-3.10(d), the RMSE value of the KNN model is lower compared to other models and we can see that the MSE of this model is almost similar to the linear regression and random forest model.

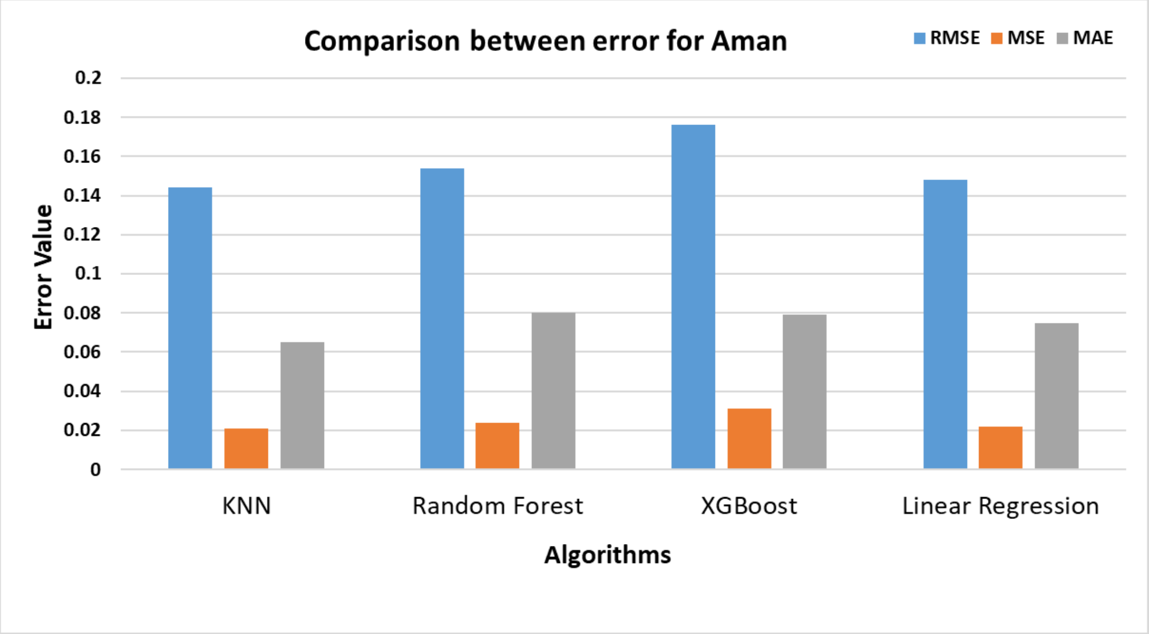


Figure-3.10.1: Comparison between Error (in tons) for Aman

3.6.6 Scatter plot for Aman

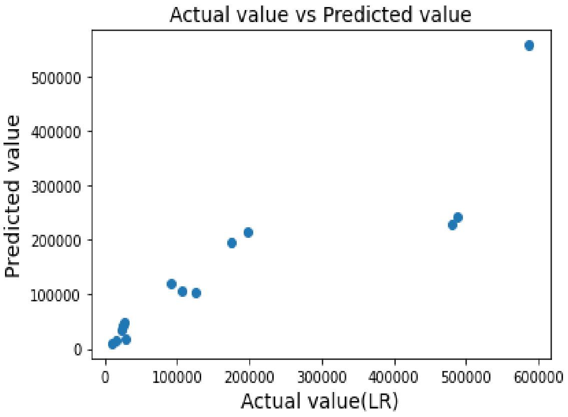


Figure-3.11(a): Scatter plot of Linear Regression

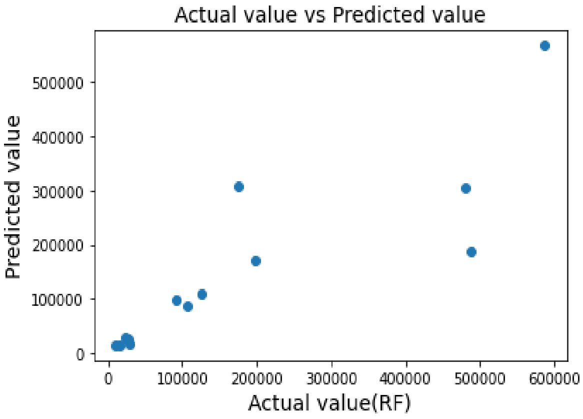


Figure-3.11(b): Scatter plot of RF

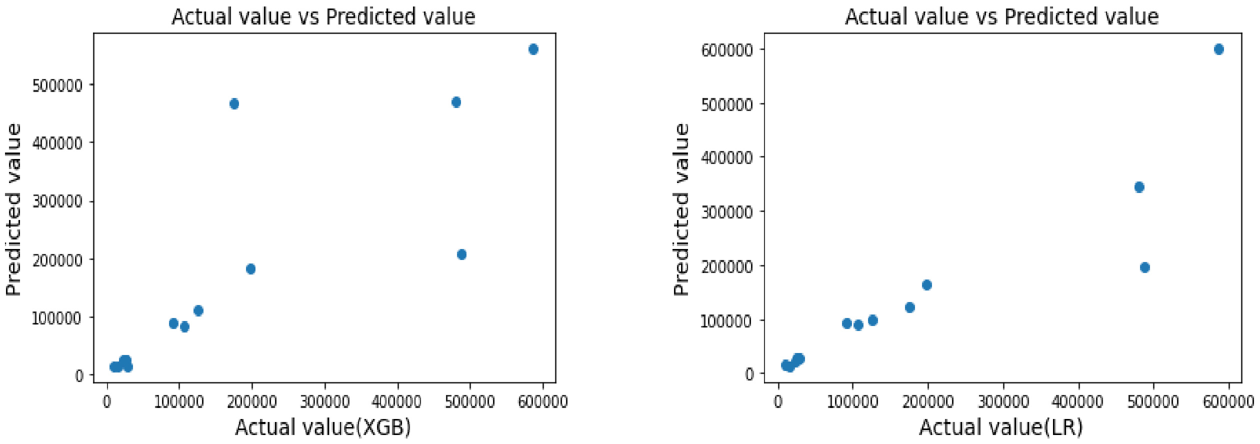


Figure-3.11(c): Scatter plot of GB

Figure-3.11(d): Scatter plot of KNN

Figure-3.11: Scatter plot for Aman

Scatter plotting helps us to explore the relationship between two variables and it's basically represented as a dotted value that shows us what will be the value in the horizontal and vertical axis. By scattering plotting, we can easily observe how the predicted and actual values are spread in the regression line.

For the above pictures, the values are closer to each other and also to the regression line. If we compare scatter plots of Aman with scatter plots of Aus then we can see that the values are more in comparison to the Aus and they are much closer. On the other hand, we have seen in the scatter plot of Aus that there were not many values available and they were also distant from each other. The above picture shows all the models have a band that can closely approximate the values.

3.6.7 Accuracy of the models for Boro

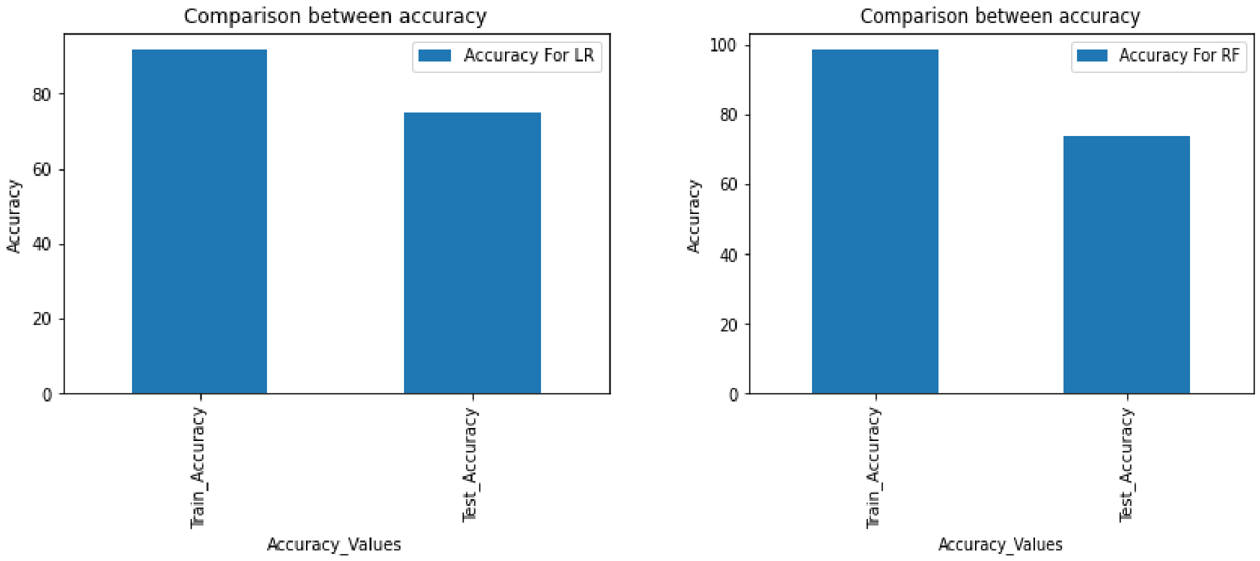


Figure-3.12(a): Accuracy of Linear regression

Figure-3.12(b): Accuracy of RF

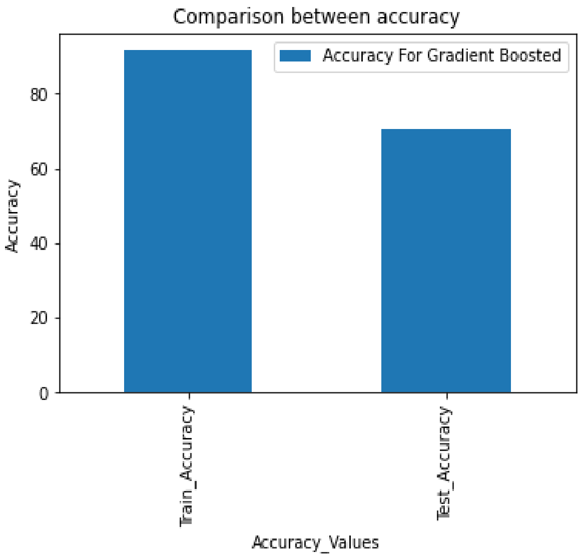


Figure-3.12(c): Accuracy of XGB

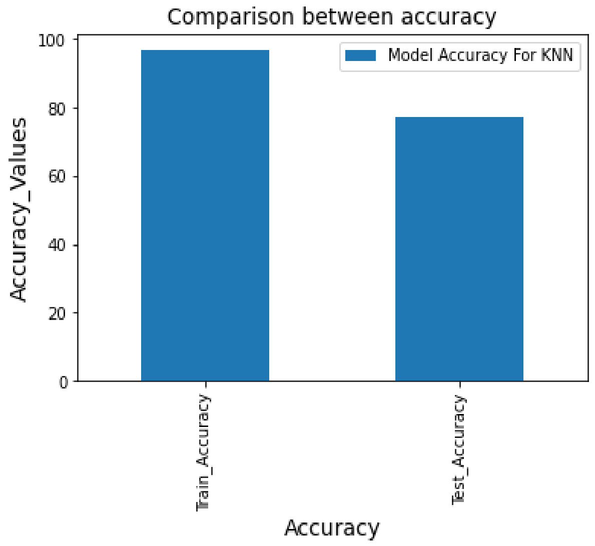


Figure-3.12(d): Accuracy of KNN

Figure-3.12: Accuracy plot for Boro

In figure-3.12(a), the accuracy of the Linear regression model for boro is shown where the training accuracy is 92% and this model has a testing accuracy of 75%.

In figure-3.12(b), the accuracy of the Random Forest model for boro is visible and this model has a training accuracy of 98% as well as the testing accuracy is 74%.

In figure-3.12(c), from the Gradient boosting model, we get the training accuracy of 92% and this model has a testing accuracy of 71%.

In figure-3.12(d), the accuracy of the KNN model for boro is displayed above where the training accuracy is 97% and this model has a testing accuracy of 77% which is highest among all algorithms.

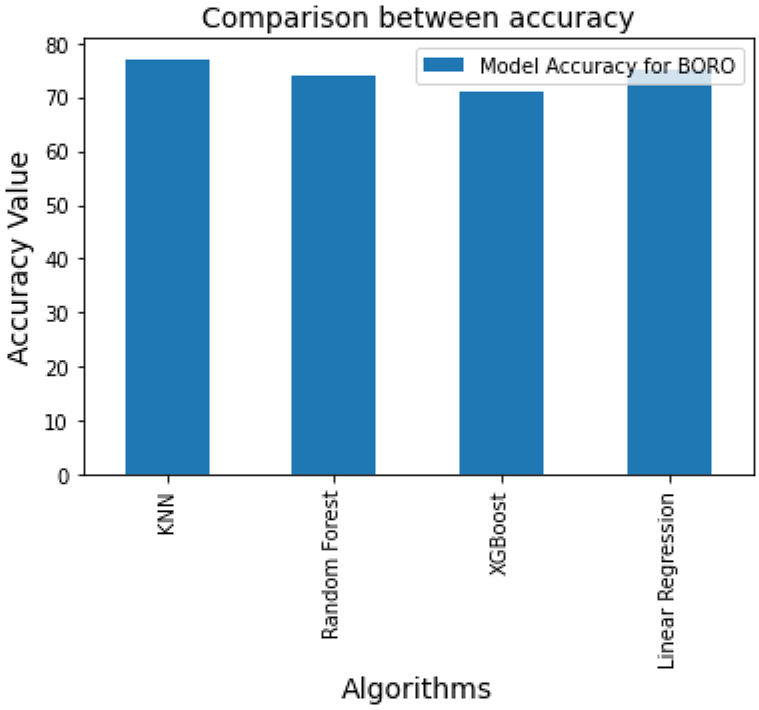


Figure-3.13: Comparison between Accuracy for Boro

3.6.8 Errors of the models for Boro

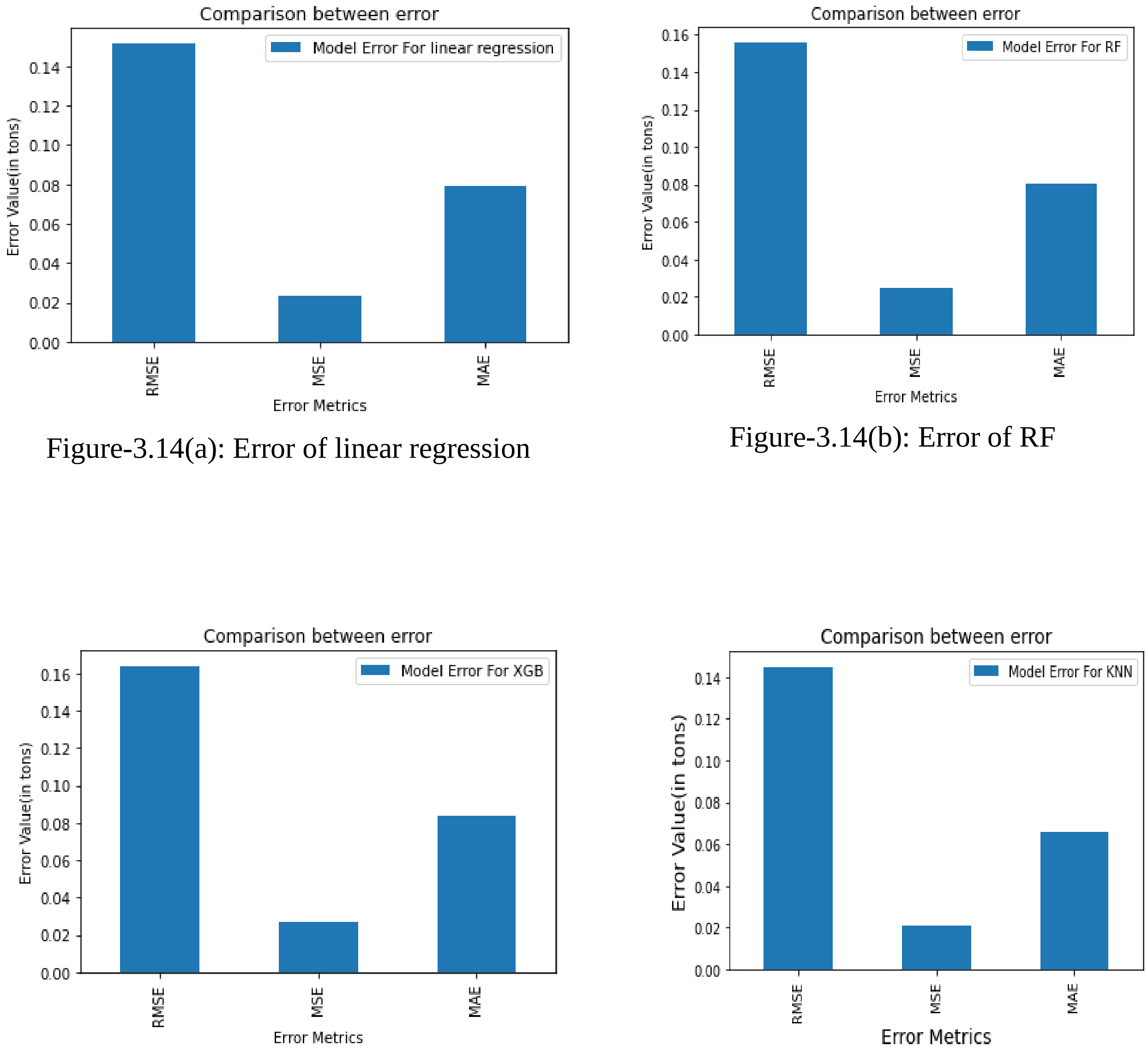


Figure-3.14(a): Error of linear regression

Figure-3.14(b): Error of RF

Figure-3.14(c): Error of XGB

Figure-3.14(d): Error of KNN

Figure-3.14: Error (in tons) for Boro

In figure-3.14(a), errors of the Linear regression model are shown where the value of RMSE is almost double than MAE. and MSE is low compared to the other error values.

In figure-3.14(b), we can see that the RMSE value of the Random forest model is much bigger compared to the mean absolute error or mean squared error of the model.

In figure-3.14(c), the RMSE value is higher than other models and therefore the accuracy is low in XGBoost model.

In figure-3.14(d), the RMSE value of the KNN model is lower compared to the other models and we can see that the MSE of this model is also low and for that reason, we got highest accuracy from this model.

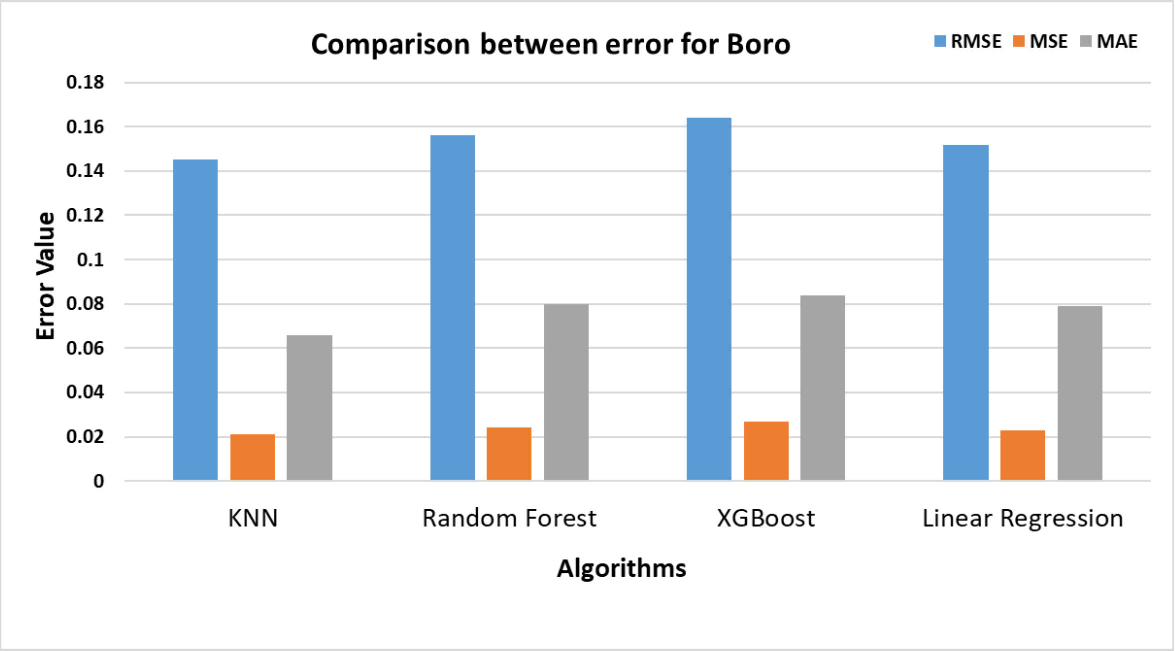


Figure-3.15: Comparison between Error (in tons) for Boro

3.6.9 Scatter plot for Boro

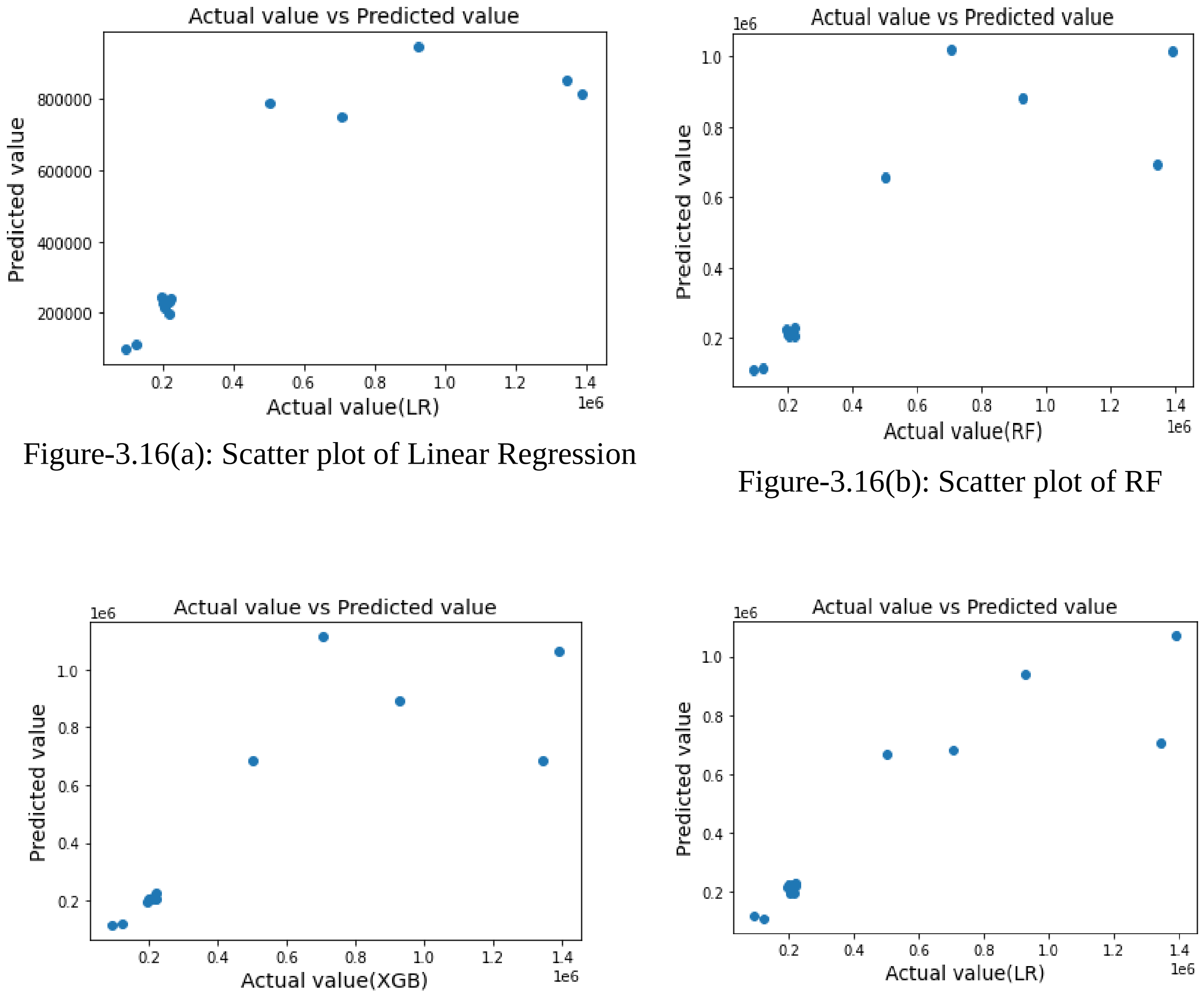


Figure-3.16(a): Scatter plot of Linear Regression

Figure-3.16(b): Scatter plot of RF

Figure-3.16(c): Scatter plot of XGB

Figure-3.16(d): Scatter plot of KNN

Figure-3.16: Scatter plot of Boro

Basically, a scatter plot allows us to observe the correlation between variables which will help us to implement the model further to make predictions. This plotting also ensures how reliable a model is. From the above model, we can observe that the values are not linearly distributed across the line rather than that they are more distant from each other, and wherein the scatter plot of the

KNN model the values are closer to the regression line compared to other models and thus we will further observe in the 3.7 section that the accuracy of KNN model is higher than other implemented models. So, to conclude, this scatter plotting also helps in predicting the result more accurately and the performance of the model also depends on that.

3.6.10 Feature importance

Feature importance of any variable can be determined by using the XGBoost algorithm and it is the way of selecting the important features which has the most impact in the yield of any crop. Feature selection can be possible by performing XGBoost algorithm. [42]
In our research, we have performed XGBoost to know which variable or parameters has the most impact on the growth of Aus, Aman, and Boro.

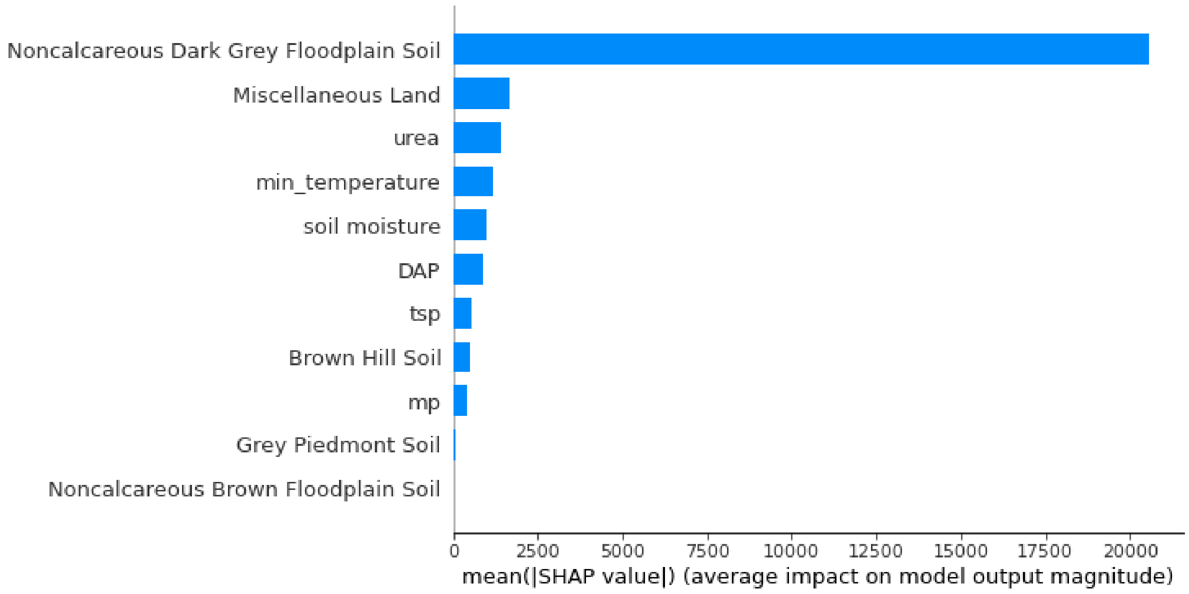


Figure-3.17: Feature importance of Aus

In figure-3.17, we can see the feature importance of Aus from which we can observe which land type, soil type, or fertilizer has the highest impact on the yield of Aus rice in Bangladesh. It is seen that non-calcareous dark grey floodplain soil has the highest impact on the Aus yield. Moreover, the land type which has the greatest impact is Miscellaneous land and the fertilizer is urea. So, to conclude, urea fertilizer, non-calcareous dark grey floodplain soil, and miscellaneous land are the dominant factors to maximize the Aus yield.

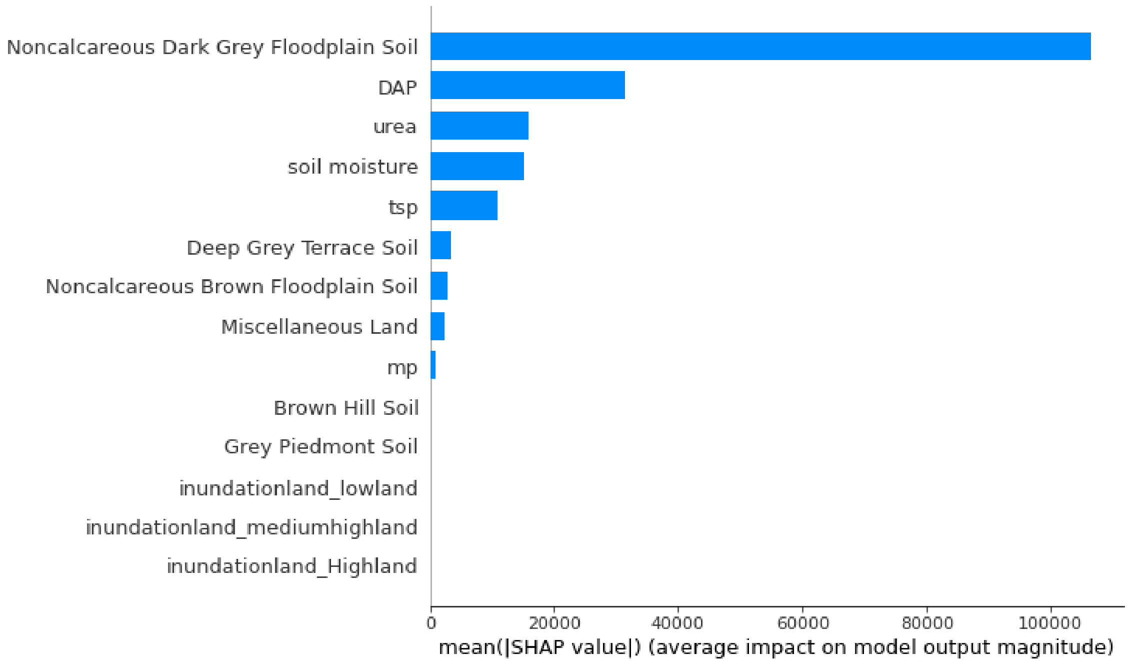


Figure-3.18: Feature importance of Aman

Figure-3.18 shows the features which are important to ensure the Aman yield in certain regions. Similar to Aus yield, non-calcareous dark grey floodplain soil has the greater importance to maximize the yield of Aman but fertilizer DAP and then urea as well as soil moisture has the highest impact on the Aman yield rather than any land type.

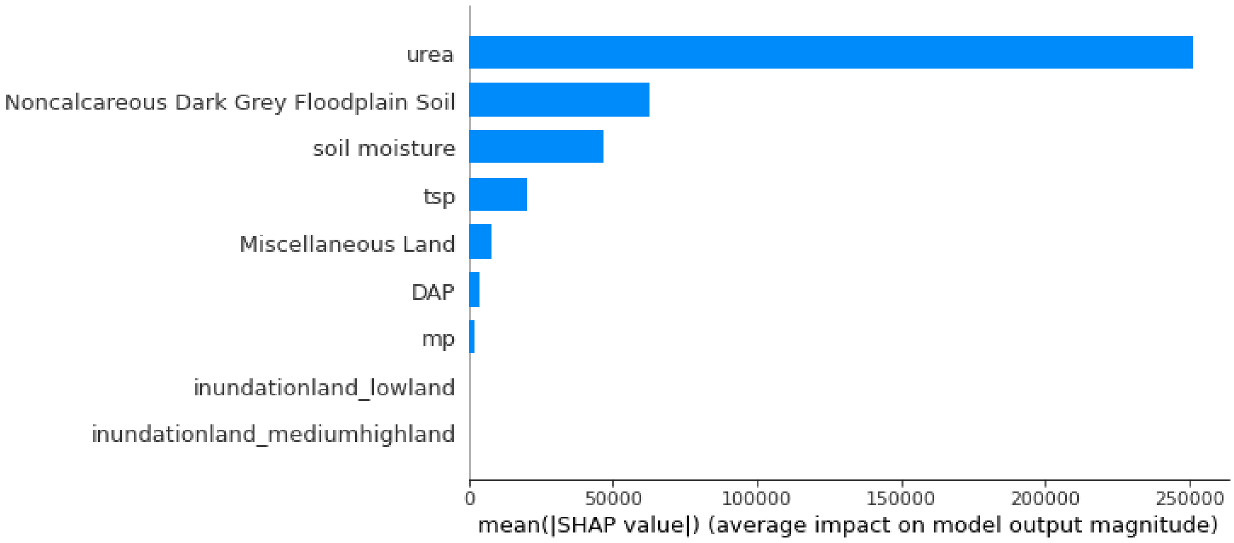


Figure-3.19: Feature importance of Boro

From figure-3.19, we can observe that fertilizer urea, non-calcareous dark grey floodplain soil, and soil moisture are the important factors to maximize the Boro yield and these factors have a larger effect on Boro yield.

3.7 Comparison among Algorithms

In this section of the paper, we have described the accuracy and errors of each model in the case of Aus, Aman, and Boro to know which model is capable of predicting more accurately for each crop. We can see from table 3.1, 3.2, and 3.3 that not all the models are giving the same accuracy. It varies from crop to crop and the accuracy or performance of the model is dependent on the correlation between the crop and other parameters.

Models	Accuracy	Root Mean Squared Error (RMSE)	Mean Absolute Error (MAE)	Mean Squared Error (MSE)
K-nearest neighbor	92%	0.110	0.052	0.012
Random Forest	90%	0.089	0.043	0.008
Gradient Boosting	90%	0.089	0.043	0.008
Linear Regression	65%	0.166	0.096	0.028

Table-3.1: Comparison of performance of different models for Aus yield prediction

Table-3.1 shows the accuracy and errors for four different algorithms to predict the Aus yield. We can see here that the KNN model has higher accuracy of 92% compared to all the other models. Also, the error factors are much lower in the case of the KNN model than the other models. The RMSE of a model can also describe the performance of the model and the lower the values of error will be, the higher will be the performance of the model. Though in KNN, the mean squared error and root mean squared error values are higher than random forest and gradient boosting algorithm, we got the highest accuracy from k-nearest neighbor algorithm. We have accuracy above 90% for three models that is KNN, gradient boosting and random forest algorithm has an accuracy of 92% and 90% for other two models respectively. Moreover, the linear regression model has the lowest accuracy in this case and Random Forest also has an accuracy of 89.66% which is almost closer to 90%. Additionally, we got lower values for mae and mse than rmse for k-nearest neighbor algorithm.

In table 3.2, we can see that the accuracy of models is much lower for Aman yield prediction than Aus yield prediction where we have a maximum accuracy of 76% and that is provided by the KNN model. So, comparing the Aus and Aman, we can say in both cases KNN is giving the higher accuracy as well as the error factors of KNN is lower than the other models. So, KNN might be a reliable model to predict the Aman yield with these agricultural factors.

Models	Accuracy	Root Mean Squared Error (RMSE)	Mean Absolute Error (MAE)	Mean Squared Error (MSE)
K-nearest neighbor	76%	0.144	0.065	0.021
Random Forest	73%	0.154	0.080	0.024
Gradient Boosting	65%	0.176	0.079	0.031
Linear Regression	75%	0.148	0.075	0.022

Table-3.2: Comparison of performance of different models for Aman yield prediction

Models	Accuracy	Root Mean Squared Error (RMSE)	Mean Absolute Error (MAE)	Mean Squared Error (MSE)
K-nearest neighbor	77%	0.145	0.066	0.021
Random Forest	74%	0.156	0.080	0.024
Gradient Boosting	71%	0.164	0.084	0.027
Linear Regression	75%	0.152	0.079	0.023

Table-3.3: Comparison of performance of different models for Boro yield prediction

From table 3.3, we can observe that we have the higher model performance for KNN algorithm which is 77% for the Boro yield prediction, and again the error factors of this model are lower which makes this model an efficient one for predicting the result more accurately.

So, from the above comparison of all models, we are able to know which model is much better or efficient to predict.

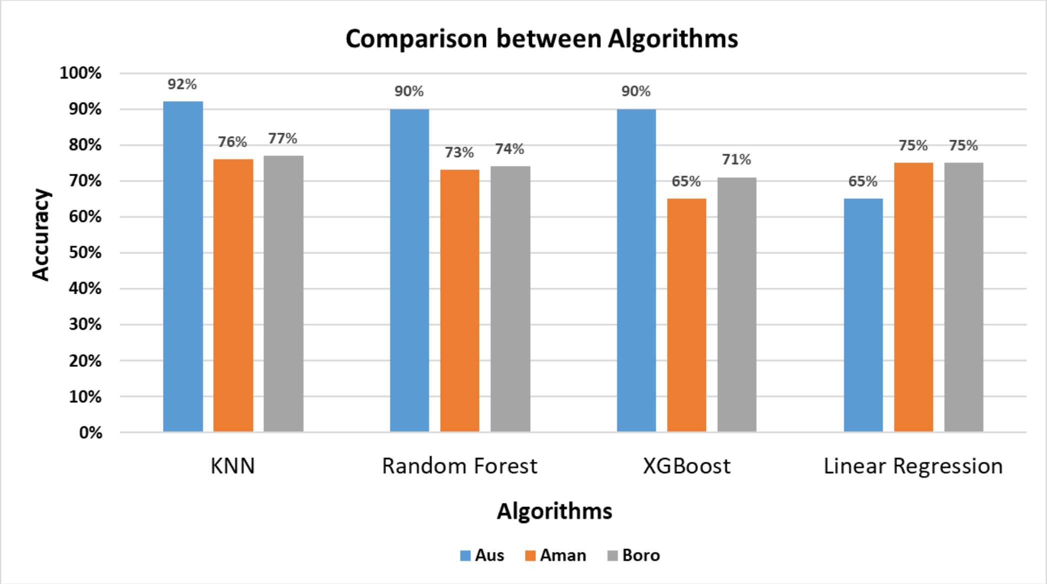


Figure-3.20: Comparison of Algorithms of Aus, Aman, and Boro

So, from the above graph, we can observe that the model accuracy for the k-nearest neighbors' model is higher than any other model which is 92% accuracy for Aus prediction, 76% for Aman, and 77% for Boro. Thus, if we try to predict the yield of Aus, Aman, and Boro in every case, KNN model is much more efficient in predicting.

3.8 Comparison with literature

Although we implemented various algorithms to analyze and predict which model has the highest model performance or accuracy, we also have some limitations in our project. To interpret the result from our analysis, we did some background research and found out some similar projects like our research topic which also worked on the crop yield using machine learning algorithms, and the main factors which impacted the yield were also agricultural data in their paper.

There is a journal of E. Manjula, where they have used k-means clustering algorithm to analyze the crop yield production and the overall accuracy was almost 90% and their error rate was low close to 15%(E. Manjula, S. Djodiltachoumy, 2017). [13] But in our case, we have low error rate for the k-nearest neighbor model which was the best one among all with an accuracy of 92% for Aus yield, 76% for Aman, and 77% for Boro.

There is a journal paper where the author has researched agricultural crop prediction using the KNN algorithm in two different districts and they found the overall accuracy of their prediction as 63.63% for the Mangalore region and 56.66% for the Kodagu region (H. K. Karthikeya et al,2020). [7] So, in our case, the accuracy for the k nearest neighbor model was higher than their result which is 92% for Aus, 76% for Aman, and 77% for Boro.

Literature			Our Findings		
Model Used	Accuracy	Error	Model Used	Accuracy	Error (RMSE)
KNN (Mangalore region) [7]	63.63%	N/A*	KNN (Aus)	92%	0.110
KNN (Kodagu region) [7]	56.66%	N/A*	KNN(Aman)	76%	0.144
K-means clustering algorithm [13]	~90%	~15%	KNN(Boro)	77%	0.145

* Not mentioned in literature.

Table 3.4: Comparison between literature and our findings

Parameters	Aman	Aus	Boro
area	x	x	x
Avg rainfall	x	x	x
Max temperature	x	x	x
Min temperature	x	0.71	x
wheat	x	x	x
potato	x	x	x
jute	x	x	x
humidity	x	x	x
storm	x	x	x
urea	0.9	0.73	0.84
tsp	0.91	0.81	0.75
mp	0.91	0.84	0.73
DAP	0.92	0.84	0.71
Inundationland highland	0.76	x	x
Inundationland mediumhighland	0.83	x	0.76
Inundationland lowland	0.83	x	0.84
Inundationland mediumlowland	x	x	x
Inundationland verylowland	x	x	x
Miscellaneous Land	0.77	0.75	0.76
Calcareous Alluvium	x	x	x
Noncalcareous Alluvium	x	x	x
Acid Basin Clay	x	x	x
Calcareous Brown Floodplain Soil	x	x	x
Calcareous Grey Floodplain Soil	x	x	x
Calcareous Dark Grey Floodplain Soil	x	x	x
Noncalcareous Grey Floodplain Soil	x	x	x

Noncalcareous Dark Grey Floodplain Soil	0.89	0.91	0.8
Peat	x	x	x
Made-Land	x	x	x
Noncalcareous Brown Floodplain Soil	0.83	0.83	x
Shallow Red-Brown Terrace Soil	x	x	x
Deep Red-Brown Terrace Soil	x	x	x
Brown Mottled Terrace Soil	x	x	x
Shallow Grey Terrace Soil	x	x	x
Deep Grey Terrace Soil	0.8	x	x
Grey Valley Soil	x	x	x
Brown Hill Soil	0.81	0.83	x
Grey Piedmont Soil	0.8	0.82	x
soil moisture	0.89	0.72	0.84

**x indicates the features (independent variable that have been dropped)

Table-3.5 Selections of the features matrix of Aman, Aus & Boro yield prediction

In table 3.5, it is shown that which features have strong correlation with Aus, Aman and Boro yield and which features we have dropped because of having poor correlation that is basically bellow 0.7. For our prediction, we have got highest accuracy by using k nearest neighbor algorithm for Aus, Aman and Boro rice which is 92% accuracy for Aus,76% for Aman and 77% for Boro and root mean squared error are 0.110, 0.144, 0.145 for Aus, Aman, and Boro respectively. We can observe that when the accuracy is 92%, error is 0.110 and when the accuracy is 76% or 77% (which is much lower than 92%), the error should be higher compared to 0.110 according to the convention. But the difference of error is not significant due to the selected features which varied for Aus, Aman and Boro. For each crop, we selected the features with highest correlation factor and dropped the features with lowest correlation which is shown in table 3.5. From the table we can conclude that there are some features like urea, tsp, dap, mp, miscellaneous land which we have used for all three crops and there are features like deep grey terrace soil that is only used for Aman as well as some features we have dropped for all three crops which can cause the problem in the error part as we are not using similar features for all the crops.

3.9 Conclusion

To conclude, in this chapter we have used agricultural factors such as fertilizer, land, many types of soil and soil moisture to predict the yield of Aus, Aman and Boro rice. We have used four types of machine learning algorithms to implement our model from which we have got the highest accuracy of 92% for Aus, 76% for Aman and 77% for Boro by using k nearest neighbor algorithm as well as the error values are comparatively low. So, in this chapter we have successfully predicted the yield of crops using machine learning algorithms with the help of agricultural features and we were able to get high accuracy for our model.

Chapter 4
Machine Learning Algorithms in Rice Yield Prediction based on Weather Factors

4.1 Introduction

Agriculture is a pillar of Bangladesh's food and economy with more than 75% of Bangladesh's population dependent on agriculture for their survival. Therefore, crop production is a very important food production in Bangladesh. [10] For this reason, we are using Machine learning to predict crop yield production and increase the production of crops. Crop yield prediction is an essential task for national and regional levels so that the decision-maker can make better decisions for rapid decision-making. [10] The farmer of Bangladesh can take good decisions on what to grow and when to grow with the help of an accurate crop yield prediction model. There are different approaches to crop yield prediction. Using Machine learning or ANNs to predict anything is not an easy task. [10] We are taking some special features that affect the growth of crop yield production so that we can predict the best way to produce crops in a very efficient way. The special features are:

- Average Temperature
- Average Moisture
- Average yearly crop yield
- Relative Humidity
- Average Bright Sunshine
- Cloud Coverage
- Average Minimum Temperature
- Average Maximum Temperature
- Wind Speed

Among the special features, the most important is the Average production of crops for the past 44 years. It is important because we correlate with other special features to determine the prediction of crops for upcoming months and years. There are different types of models to predict crop yield production. These are: [1]

1. k-Nearest Neighbor
2. Decision Trees and Random forest
3. Support Vector Machine
4. Neural Networks
5. Linear Regression
6. Gradient Boosting
7. Logistic Regression and so on

Among all the models we are using the Linear Regression Model, k-Nearest Neighbor, Gradient Boosting and Random forest to predict crop yield production. Besides, there are a lot of research papers like us and we are taking their help from their paper too. All the papers and research are very vital for us to build the ANNs system to predict crop production.

These papers present a combination of agricultural mapping and monitoring of different locations of Bangladesh with crop growth and yield prediction. Moreover, there are also many problems the researcher has faced while making the ANNs system for predicting the Crop yield production and we are taking these problems into serious consideration to make our prediction as good as possible. [10]

Bangladesh is agricultural land and rice is the staple food for all the people of Bangladesh. About 75% of the people of the country are professional farmers. Hence, Rice or crop production is very important for our food and nutrient values. Rice or crops are easily produced on our land because the land of Bangladesh is very fertile. This is because over 70% of the land of Bangladesh is arable which is able to support farming and crops. [10] Furthermore, the Topical temperature and well water climate make it perfect for farming. Most importantly Bangladesh has a lot of rivers and river pathways are very dense. Therefore, Bangladesh is also known as River Country. There is a lot of rice produced in Bangladesh. These are:

- Aus
- Amon
- Boro
- HYV
- Pajam

Day by day farmers are trying new types of methods to produce more crops. For example: making a crossbreed, planting a hybrid, etc. [36] Recently, the agricultural sector of Bangladesh has more type-made crops than before. Many renowned researchers and university students and teachers are using Machine learning to predict and produce crops in a small place. Many of them did make ANNs or Machine learning systems based on crop yield production. This system is made which is based on special features or data and given below: [39]

- Temperature
- Moisture
- production of the crop for past 44 year
- Soil type
- Rainfall
- Irrigation system
- Soil Moisture.

Furthermore, these special features are taken while producing crops so these data are very important. Therefore, with the help of all this data or special features, we can make ANNs or Machine learning systems that can allow us to reproduce more crops in the upcoming future.

4.2 Related Work

Machine learning is one of the most important and vital subjects for the upcoming development of every sector and future generation. There are many researchers, teachers, and students who are doing research in this Crop yield prediction using machine learning.

There is a good article titled “Predicting rice yield for Bangladesh by exploiting weather conditions" done by Md. Akter Hossain and his partners.[43] They have used machine learning for making predictions on crop yield predictions based on climate change and the condition of Bangladesh. [43] They have utilized WPSRY (Weather-based Prediction System for Rice Yield), Neural Networks (NN), and Support Vector Regression (SVR). [43] With the help of WPSRY, they have predicted the rice yield in different regions of Bangladesh and this model is built to predict the weather parameter by applying a Neural network System. [43] After that by Applying Support Vector Regression (SVR) which utilizes inputs predicted weather from Neural network and current agricultural data, they estimate the rice yield. [43]

In the article “Sustainable Rice Production Analysis and Forecasting Rice Yield Based on Weather Circumstances Using Data Mining Techniques for Bangladesh”, Mohammed Mahmudur Rahman and his partners have analyzed sustainable rice production based on weather circumstances by applying data mining techniques which is also a part of Machine learning. [44] They have used Multiple Linear Regression, Support Vector Machine, and Random Forest methods to collect data for selected regions of Bangladesh by the system data mining. [44] In this way, they have got a large amount of data for getting more correct and accurate Rice Production. After that, they have analyzed the Sustainable Rice Yield Based on Weather Circumstances. [44]

There is an article named “Integrated phenology and climate in rice yields prediction using machine learning methods” which is a very good article. [45] Rice is a very demanding and staple cereal crop all around the world so they have focused on increasing crop yield prediction using machine learning. [45] In this article, there is the use of (MLR) multiple linear regression, backpropagation neural network (BP), support vector machine (SVM), and random forest (RF). [45] Here, it is shown that the ML method is more precise and accurate than MLR. Furthermore, there are combinations of integrated phenology, climate change during the season, and geographical data. All these are used to make predictions of getting more rice yields. [45]

During our research, we have found a good article “Crop yield prediction using machine learning: A systematic literature review” which is a review of many articles based on crop yield prediction using machine learning or deep learning. [10] In this article, there are a lot of models which are made based on unique and different features. Chlingaryan and Sukkarieh have done research based on nitrogen-based crop yield prediction using machine learning [Chlingaryan et al., 2018]. [10] Mayuri and Priya have performed machine learning based on image processing to identify crop diseases [Mayuri and Priya]. [10] There are more of these types of topics and articles that are reviewed and discussed here. All these articles help us to understand that machine learning is making a great influence on the agriculture sector.

4.3 Overview of Dataset

The dataset used in the machine learning models implemented is a customized amalgamation of two different datasets. The first one is 65 Years of Weather Data Bangladesh (1948 - 2013). It consisted of integer-based weather factors such as maximum temperature, minimum temperature, rainfall, relative humidity, wind speed, cloud coverage, bright sunshine, station names, x correlation, y correlation, latitude, longitude, altitude, and period. The data was substantially detailed with monthly information of over 65 years. [46] It was sourced from an open-source data repository and machine learning platform called Kaggle. The second dataset was obtained from the Bangladesh Bureau of Statistics website. It is called 44 Years of Major Crops - a dataset comprising yearly district-wise comprehensive data of different types of rice yields such as Aman, Boro and Aus. [47] Due to a difference in years and districts of data available of weather and yield we extracted data of overlapping years and districts. Additionally, since the yield data was available on a yearly basis and the weather data on monthly basis, we accumulated the weather data and averaged the values for yearly input. This resulted in a dataset that showed yearly, district-wise weather information against the total yield of that year of Aman, Aus and Boro rice.

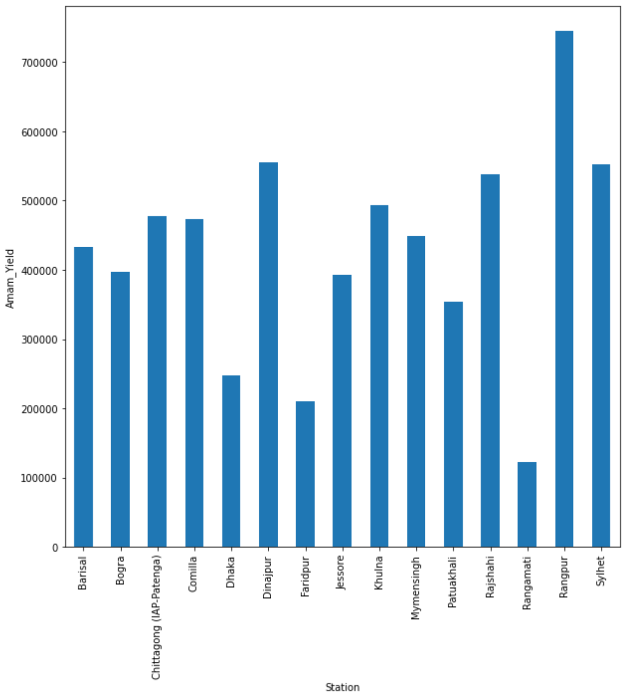


Figure 4.1(a): District wise Aman rice production

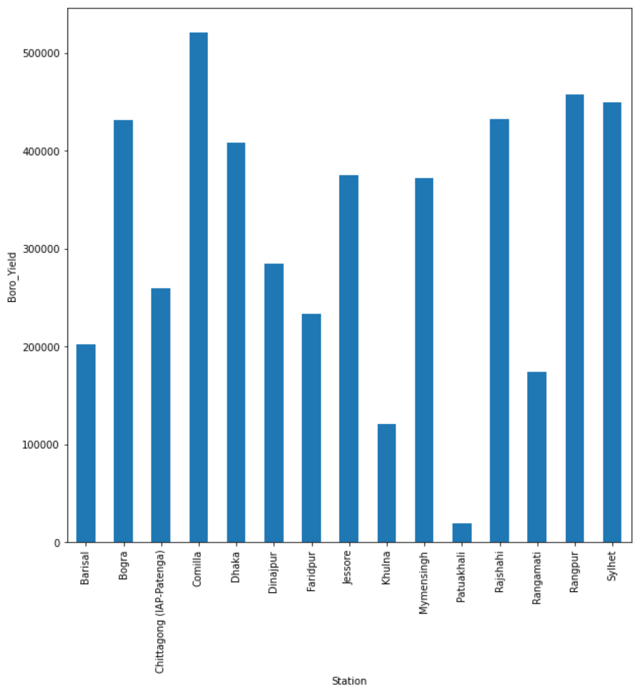


Figure 4.1(b): District wise Boro rice production

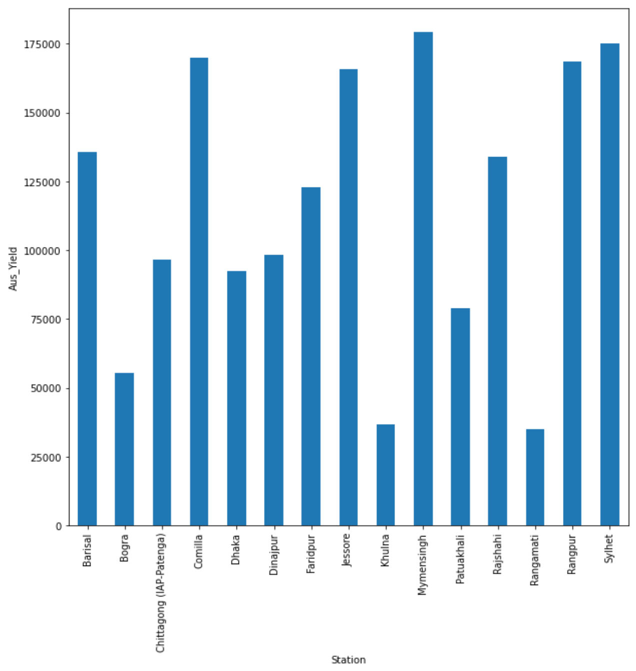


Figure 4.1(c): District wise Aus rice production

Figure 4.1: District wise Aman, Boro, and Aus production

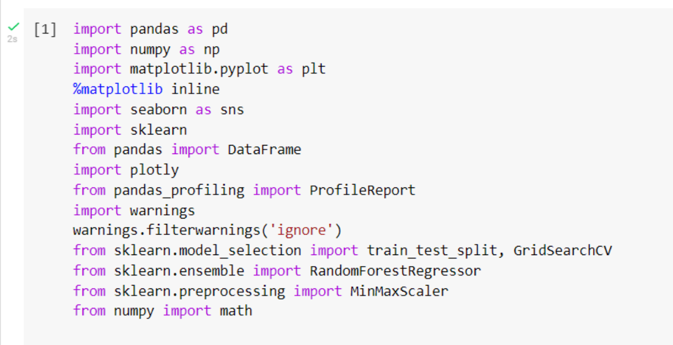
4.4 General Procedure

Here we are going to describe the algorithm and model which are used in Rice Yield based on Weather Factors.

4.4.1 Data preprocessing

Importing Library:

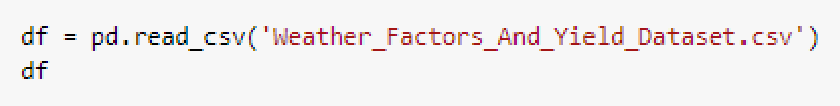
A total of 7 python-based modules and libraries were used for the data analysis and interpretation for different purposes as explained in chapter 3 and similar ones are used here.



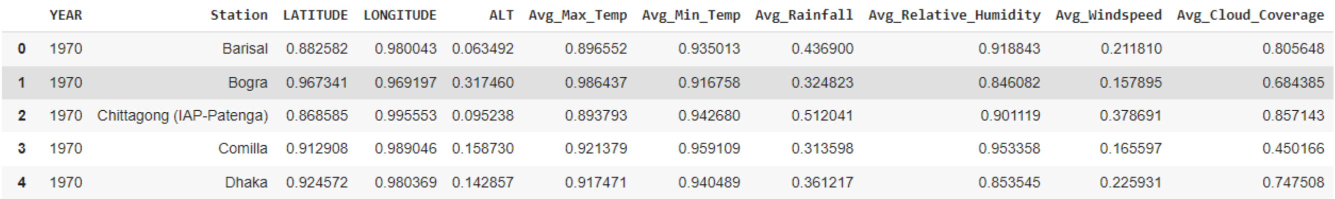
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import sklearn
from pandas import DataFrame
import plotly
from pandas_profiling import ProfileReport
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import MinMaxScaler
from numpy import math
```

Importing dataset:

The google colab platform allows direct uploading of comma separated values or .csv files and we used pd.read_csv() to import the dataset. [41]



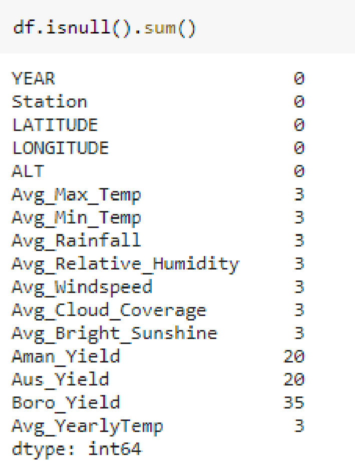
```python
df = pd.read_csv('Weather_Factors_And_Yield_Dataset.csv')
df
```



	YEAR	Station	LATITUDE	LONGITUDE	ALT	Avg_Max_Temp	Avg_Min_Temp	Avg_Rainfall	Avg_Relative_Humidity	Avg_Windspeed	Avg_Cloud_Coverage
0	1970	Barisal	0.882582	0.980043	0.063492	0.896552	0.935013	0.436900	0.918843	0.211810	0.805648
1	1970	Bogra	0.967341	0.969197	0.317460	0.986437	0.916758	0.324823	0.846082	0.157895	0.684385
2	1970	Chittagong (IAP-Patenga)	0.868585	0.995553	0.095238	0.893793	0.942680	0.512041	0.901119	0.378691	0.857143
3	1970	Comilla	0.912908	0.989046	0.158730	0.921379	0.959109	0.313598	0.953358	0.165597	0.450166
4	1970	Dhaka	0.924572	0.980369	0.142857	0.917471	0.940489	0.361217	0.853545	0.225931	0.747508

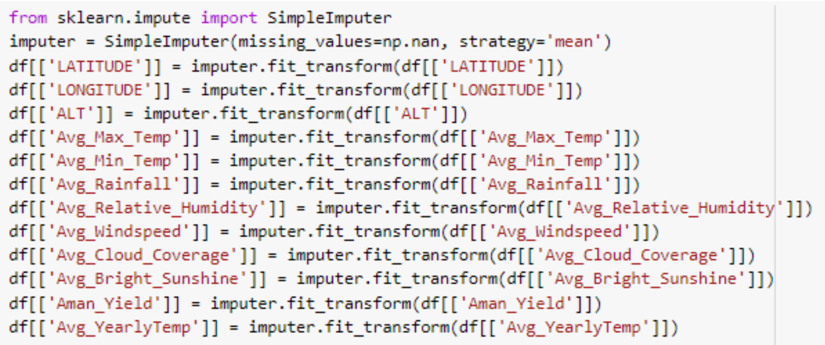
Checking Missing value

The sklearn.impute library from the Simple Imputer module was used to make up for missing values in the dataset using mean as strategy i.e. the missing values would be replaced by the average of the column.



```python
df.isnull().sum()
```

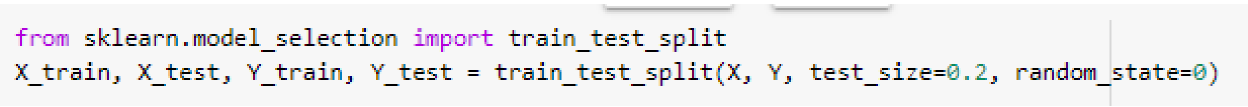
```
YEAR                      0
Station                   0
LATITUDE                  0
LONGITUDE                 0
ALT                       0
Avg_Max_Temp              3
Avg_Min_Temp              3
Avg_Rainfall              3
Avg_Relative_Humidity     3
Avg_Windspeed             3
Avg_Cloud_Coverage        3
Avg_Bright_Sunshine       3
Aman_Yield               20
Aus_Yield                20
Boro_Yield               35
Avg_YearlyTemp            3
dtype: int64
```



```python
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values=np.nan, strategy='mean')
df[['LATITUDE']] = imputer.fit_transform(df[['LATITUDE']])
df[['LONGITUDE']] = imputer.fit_transform(df[['LONGITUDE']])
df[['ALT']] = imputer.fit_transform(df[['ALT']])
df[['Avg_Max_Temp']] = imputer.fit_transform(df[['Avg_Max_Temp']])
df[['Avg_Min_Temp']] = imputer.fit_transform(df[['Avg_Min_Temp']])
df[['Avg_Rainfall']] = imputer.fit_transform(df[['Avg_Rainfall']])
df[['Avg_Relative_Humidity']] = imputer.fit_transform(df[['Avg_Relative_Humidity']])
df[['Avg_Windspeed']] = imputer.fit_transform(df[['Avg_Windspeed']])
df[['Avg_Cloud_Coverage']] = imputer.fit_transform(df[['Avg_Cloud_Coverage']])
df[['Avg_Bright_Sunshine']] = imputer.fit_transform(df[['Avg_Bright_Sunshine']])
df[['Aman_Yield']] = imputer.fit_transform(df[['Aman_Yield']])
df[['Avg_YearlyTemp']] = imputer.fit_transform(df[['Avg_YearlyTemp']])
```

Training and Testing Data

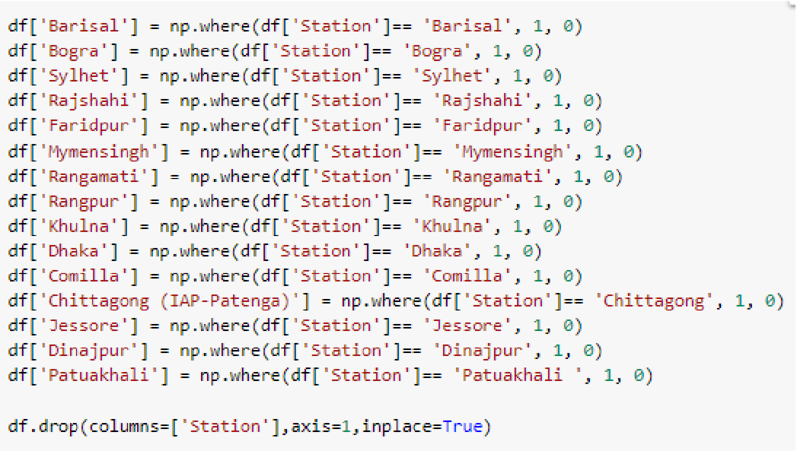
The dataset needed to be split into two parts i.e. training and testing data. It was done using the train_test_split function from the sklearn.model_selection module. 20% of the data was used for testing while 80% was used for training it.



```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=0)
```

Handling Categorical data

Categorical data shows the encoding of string information to binary ones and zeros across columns of districts.



```
df['Barisal'] = np.where(df['Station']== 'Barisal', 1, 0)
df['Bogra'] = np.where(df['Station']== 'Bogra', 1, 0)
df['Sylhet'] = np.where(df['Station']== 'Sylhet', 1, 0)
df['Rajshahi'] = np.where(df['Station']== 'Rajshahi', 1, 0)
df['Faridpur'] = np.where(df['Station']== 'Faridpur', 1, 0)
df['Mymensingh'] = np.where(df['Station']== 'Mymensingh', 1, 0)
df['Rangamati'] = np.where(df['Station']== 'Rangamati', 1, 0)
df['Rangpur'] = np.where(df['Station']== 'Rangpur', 1, 0)
df['Khulna'] = np.where(df['Station']== 'Khulna', 1, 0)
df['Dhaka'] = np.where(df['Station']== 'Dhaka', 1, 0)
df['Comilla'] = np.where(df['Station']== 'Comilla', 1, 0)
df['Chittagong (IAP-Patenga)'] = np.where(df['Station']== 'Chittagong', 1, 0)
df['Jessore'] = np.where(df['Station']== 'Jessore', 1, 0)
df['Dinajpur'] = np.where(df['Station']== 'Dinajpur', 1, 0)
df['Patuakhali'] = np.where(df['Station']== 'Patuakhali ', 1, 0)

df.drop(columns=['Station'],axis=1,inplace=True)
```

4.5 Graphs

4.5.1 Non-Correlation Graph for Aman

The non-correlation graph shows which independent variables have no correlation with the dependent variable i.e. the yield of Aman rice.

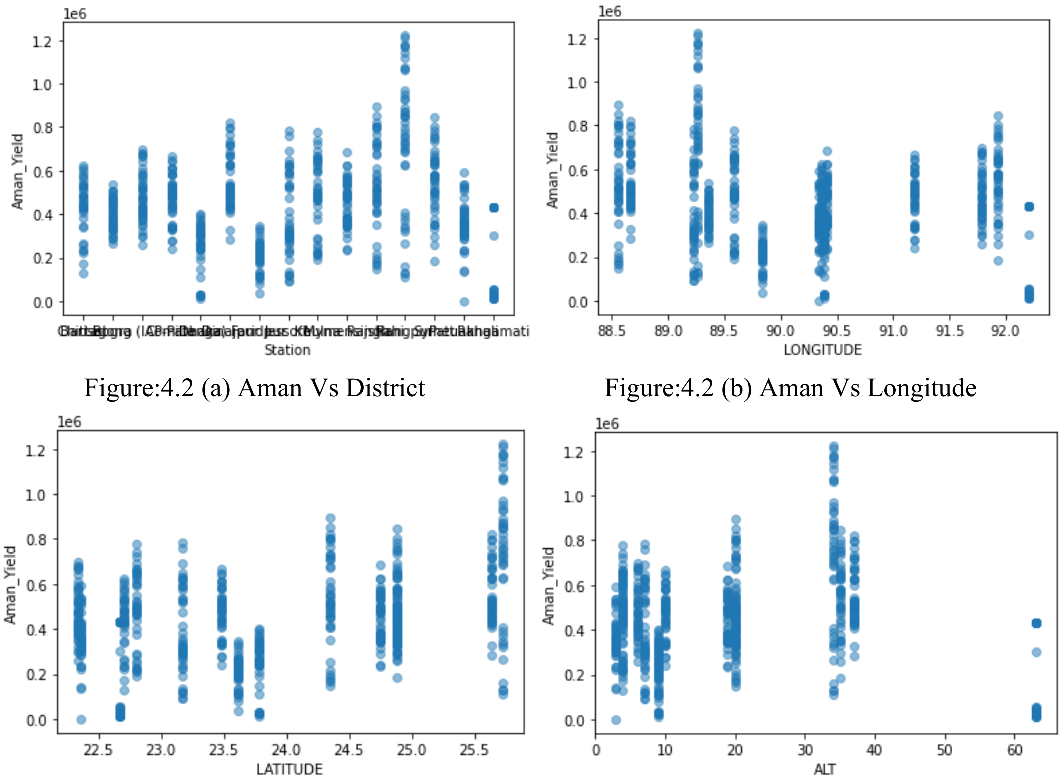


Figure:4.2 (a) Aman Vs District

Figure:4.2 (b) Aman Vs Longitude

Figure:4.2 (c) Aman Vs Latitude

Figure:4.2 (d) Aman Vs Altitude

Figure: 4.2 Non-Correlation Graph for Aman

4.5.2 Correlation Graph for Aman

The correlation graph shows which independent variables have a correlation with the dependent variable i.e. the yield of Aman rice.

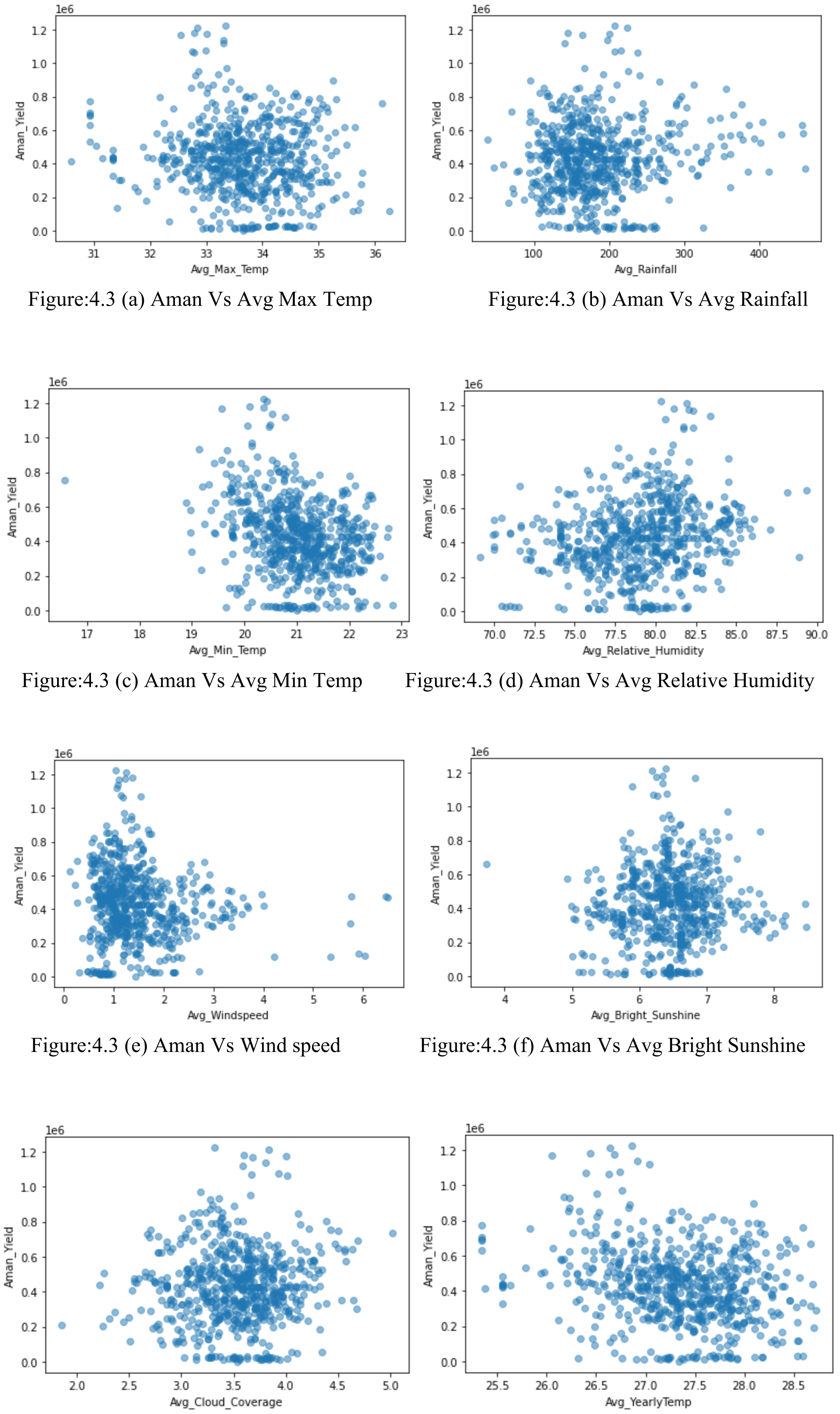


Figure:4.3 (a) Aman Vs Avg Max Temp

Figure:4.3 (b) Aman Vs Avg Rainfall

Figure:4.3 (c) Aman Vs Avg Min Temp

Figure:4.3 (d) Aman Vs Avg Relative Humidity

Figure:4.3 (e) Aman Vs Wind speed

Figure:4.3 (f) Aman Vs Avg Bright Sunshine

Figure:4.3 (g) Aman Vs Avg Cloud Coverage

Figure:4.3 (h) Aman Vs Avg Yearly Temp

Figure:4.3 Correlation Graph for Aman

4.5.3 Non-Correlation Graph for Aus

The non-correlation graph shows which independent variables have no correlation with the dependent variable i.e. the yield of Aman rice.

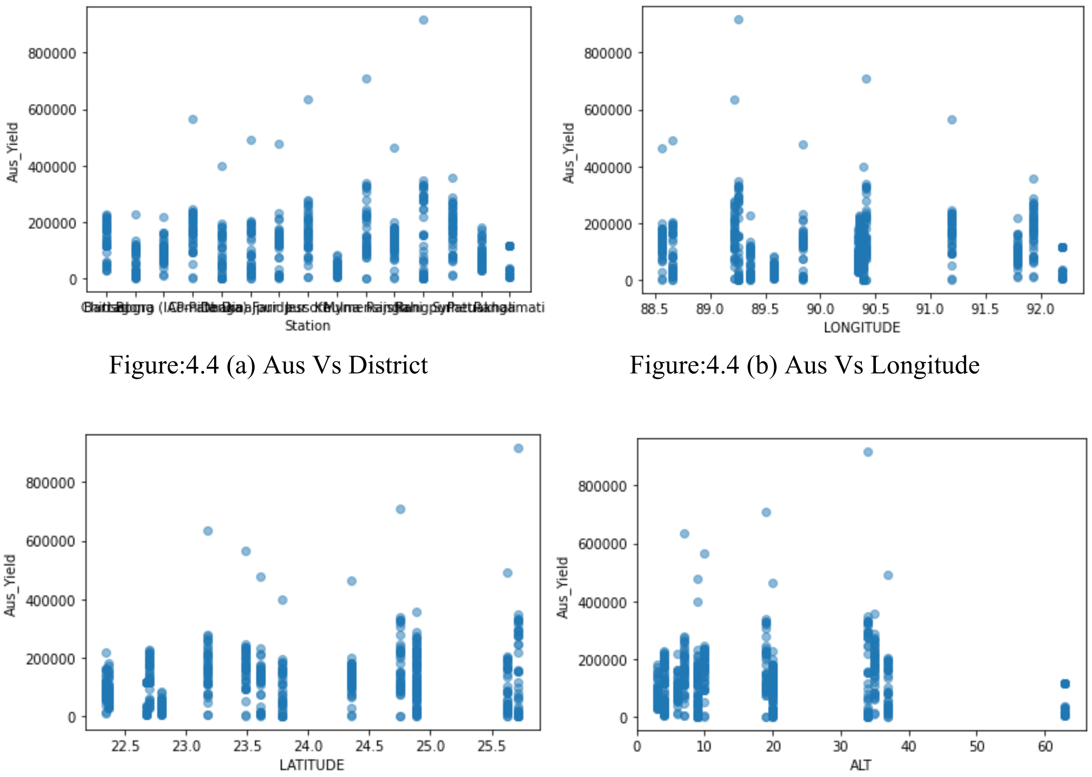


Figure:4.4 (a) Aus Vs District

Figure:4.4 (b) Aus Vs Longitude

Figure:4.4 (c) Aus Vs Latitude

Figure:4.4 (d) Aus Vs Altitude

Figure: 4.4 Non-Correlation Graph for Aus

4.5.4 Correlation Graph for Aus

The correlation graph shows which independent variables have a correlation with the dependent variable

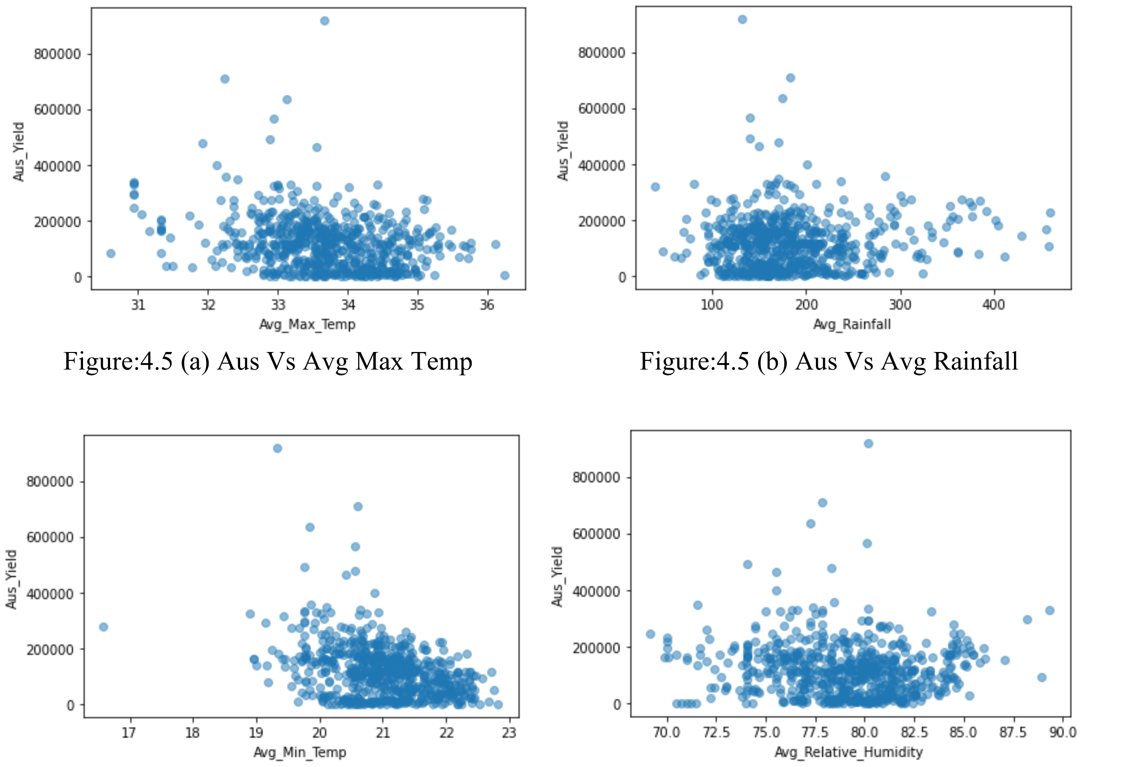


Figure:4.5 (a) Aus Vs Avg Max Temp

Figure:4.5 (b) Aus Vs Avg Rainfall

Figure:4.5 (c) Aus Vs Avg Min Temp

Figure:4.5 (d) Aus Vs Avg Relative Humidity

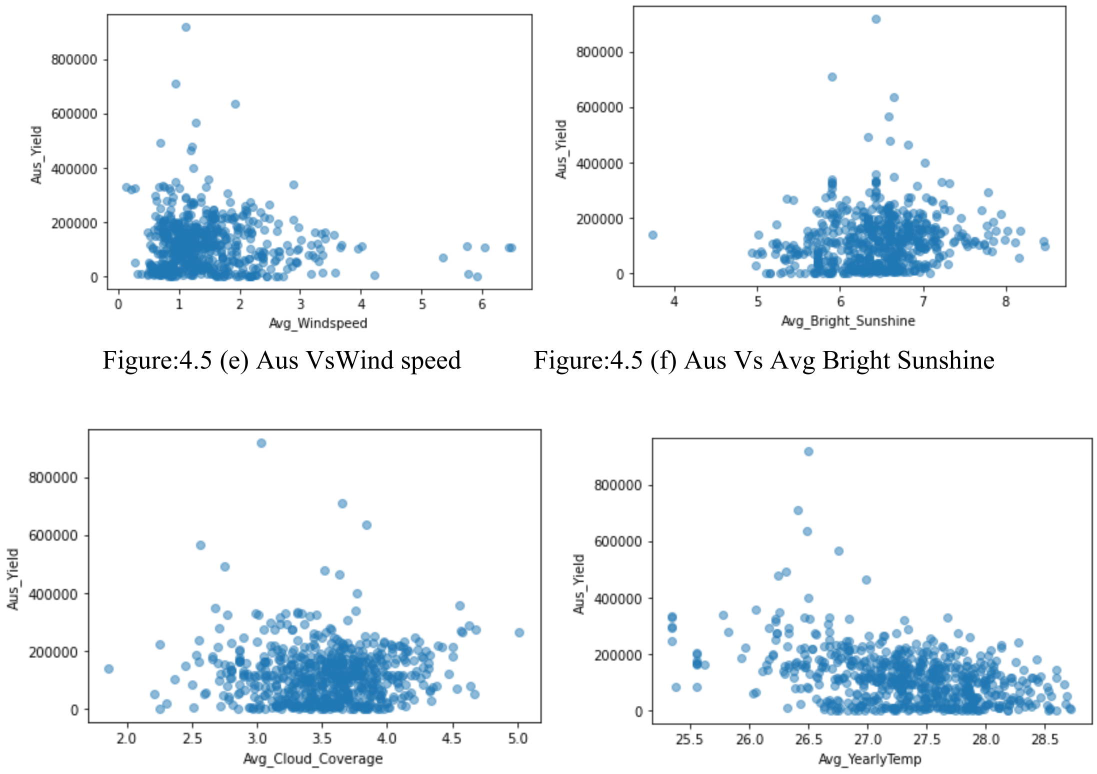


Figure:4.5 (e) Aus VsWind speed

Figure:4.5 (f) Aus Vs Avg Bright Sunshine

Figure:4.5 (g) Aus Vs Avg Cloud Coverage

Figure:4.5 (h) Aus Vs Avg Yearly Temp

Figure:4.5 Correlation Graph for Aus

4.5.5 Non-Correlation Graph for Boro

The non-correlation graph shows which independent variables have no correlation with the dependent variable i.e the yield of Boro rice.

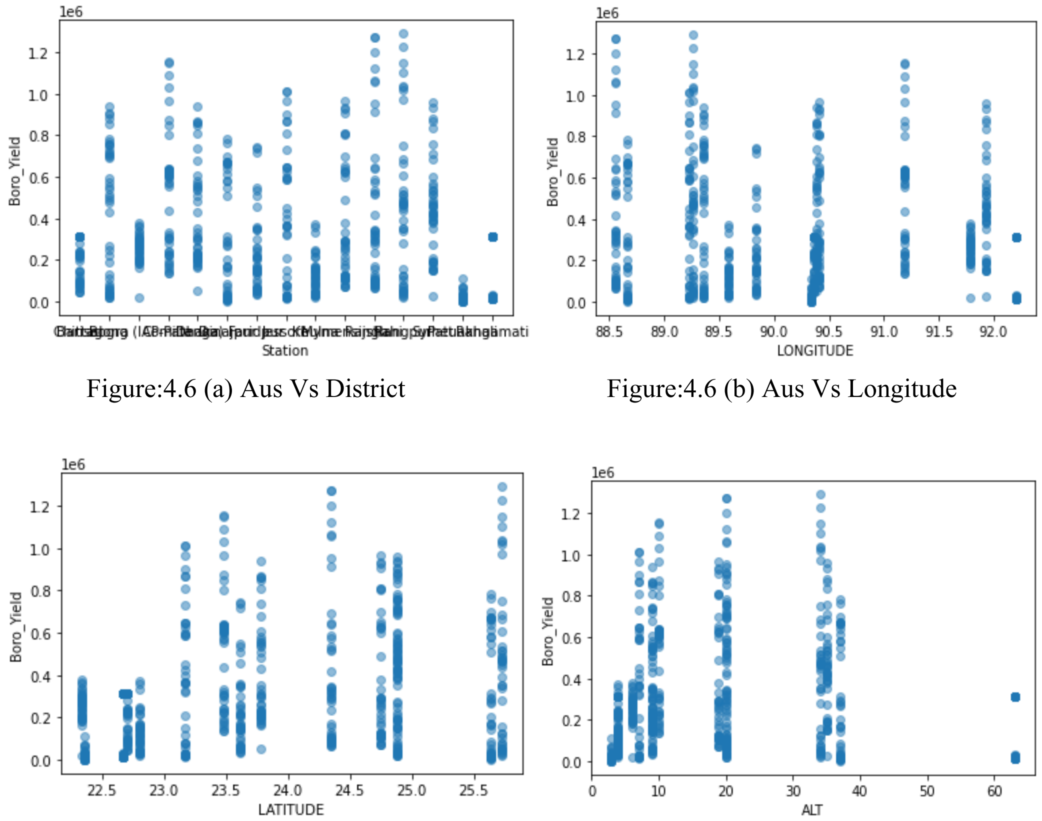


Figure:4.6 (a) Aus Vs District

Figure:4.6 (b) Aus Vs Longitude

Figure:4.6 (c) Aus Vs Latitude

Figure:4.6 (d) Aus Vs Altitude

Figure: 4.6 Non-Correlation Graph for Boro

4.5.6 Correlation Graph for Boro

The correlation graph shows which independent variables have correlation with the dependent variable i.e. the yield of Aman rice.

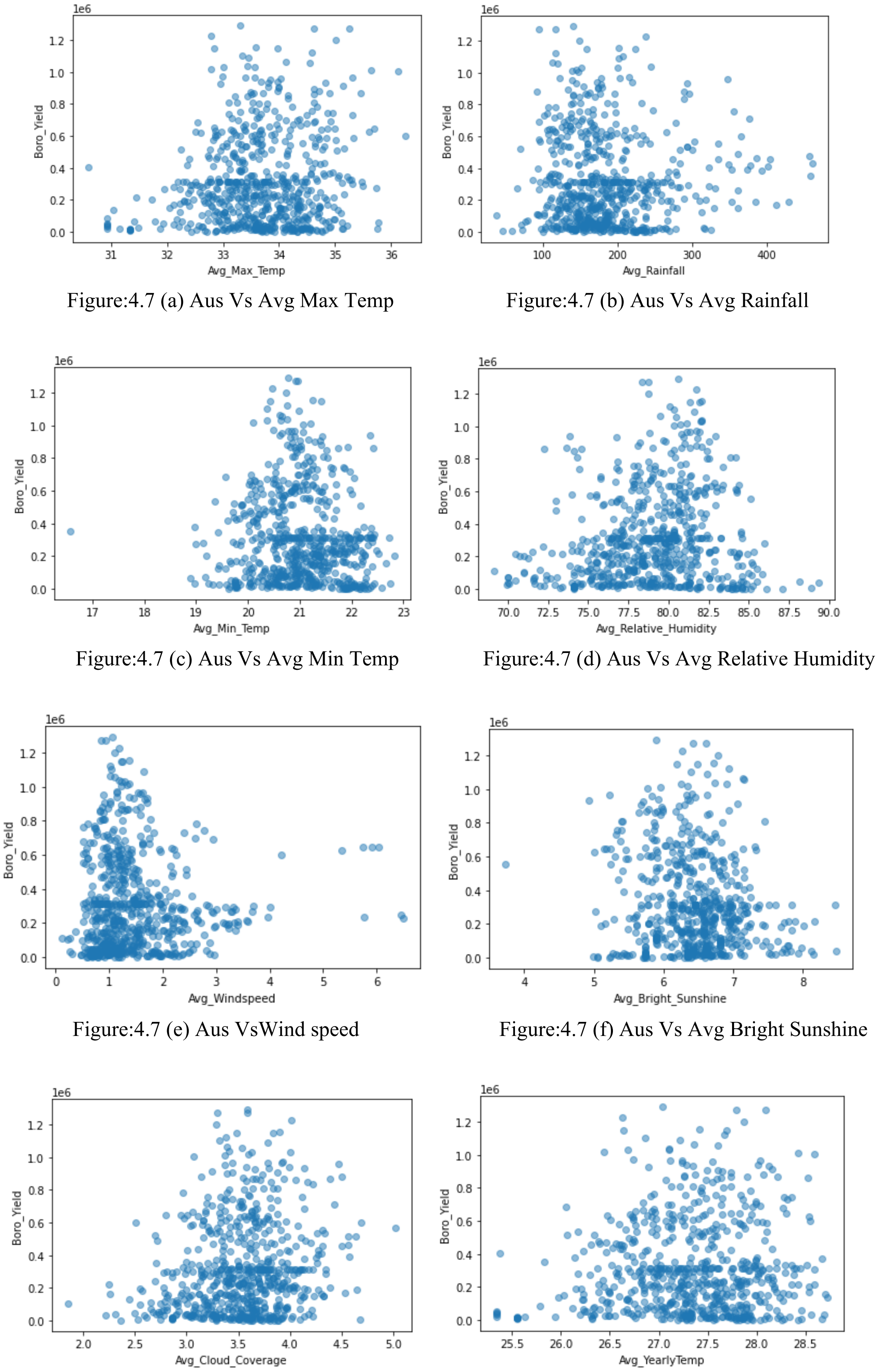


Figure:4.7 (a) Aus Vs Avg Max Temp

Figure:4.7 (b) Aus Vs Avg Rainfall

Figure:4.7 (c) Aus Vs Avg Min Temp

Figure:4.7 (d) Aus Vs Avg Relative Humidity

Figure:4.7 (e) Aus VsWind speed

Figure:4.7 (f) Aus Vs Avg Bright Sunshine

Figure:4.7 (g) Aus Vs Avg Cloud Coverage

Figure:4.7 (h) Aus Vs Avg Yearly Temp

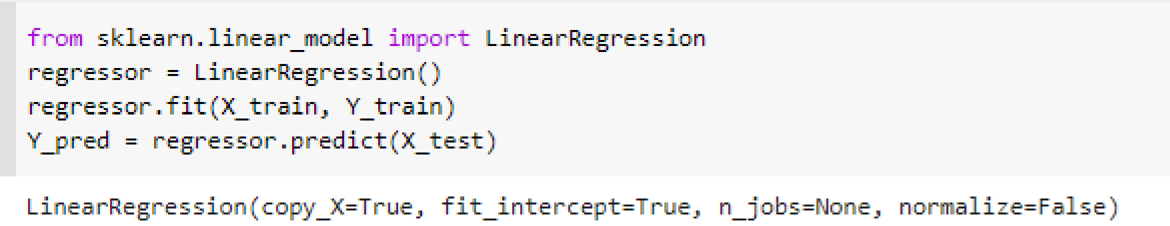
Figure:4.7 Correlation Graph for Boro

4.6 Model implementation

The project uses four different Machine Learning algorithms such as: Linear Regression, Random Forest, K Nearest Neighbor, and XGBoost. For each of the algorithms we have gotten an accuracy ready for training and testing data as well as Root mean squared error, Mean Absolute Error and Mean Squared Error.

4.6.1 Linear Regression Model

Linear regression is a one-dimensional approach of modeling a relationship between a scalar and one or more scalar variables. In this instance, we are using the linear regression library from the sklearn.linear_model module to generate regressor.coef_ (coefficient) and regressor.intercept_ (interception point). We also generated error metrics using mean squared error and mean absolute error libraries from sklearn.metrics module and accuracy from r2_score library. [14]

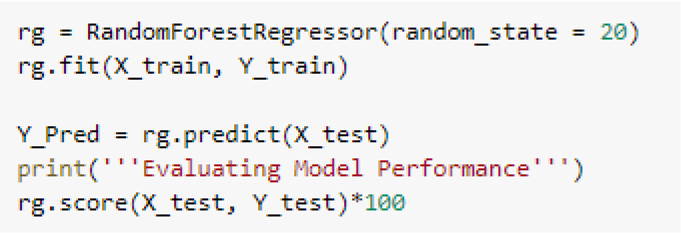


```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, Y_train)
Y_pred = regressor.predict(X_test)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

4.6.2 Random Forest

Random forests or random decision forests is a collection of learning methods for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. As for our second model, we called the RandomForestRegressor function and applied mean squared error, mean absolute error and root mean squared error metrics to evaluate the error margins. [14]

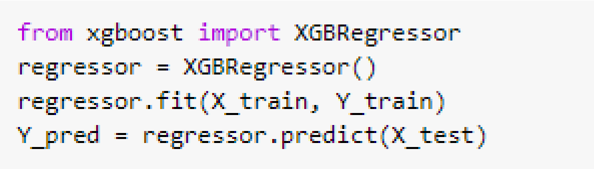


```
rg = RandomForestRegressor(random_state = 20)
rg.fit(X_train, Y_train)

Y_Pred = rg.predict(X_test)
print('''Evaluating Model Performance''')
rg.score(X_test, Y_test)*100
```

4.6.3 XGBoost Algorithm

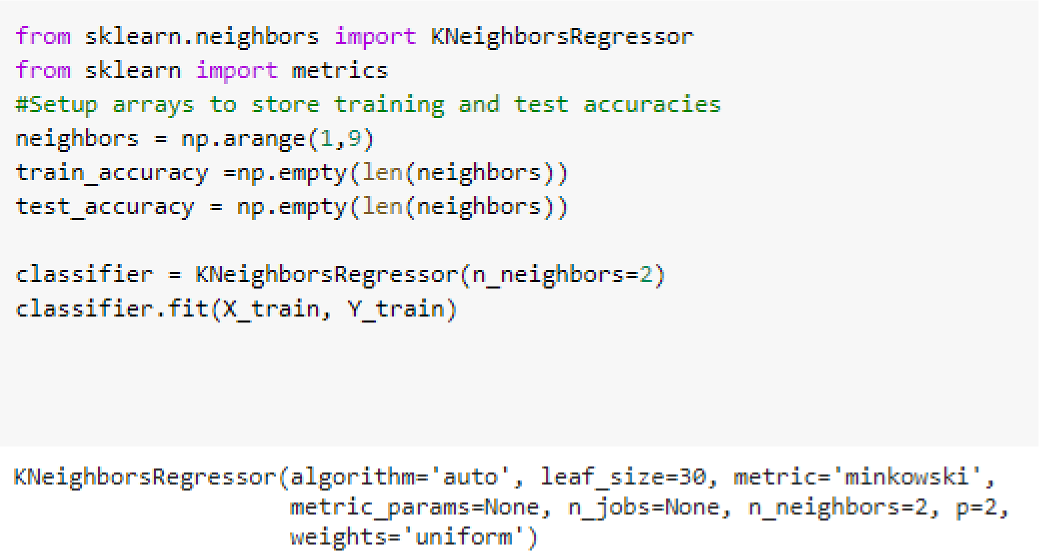
XGBoost is an open-source computer program library which gives the slope boosting system for C++, Java, Python, R, and Julia. For our third show, we utilized XGBRegressor() for boosting the existing relapse demonstration and once more connected cruel supreme blunder, cruel squared blunder, and root cruel squared blunder for calculating mistake margins. [8]



```
from xgboost import XGBRegressor
regressor = XGBRegressor()
regressor.fit(X_train, Y_train)
Y_pred = regressor.predict(X_test)
```

4.6.4 K Nearest Neighbor

KNN or K nearest neighbor may be a machine learning method and calculation that can be utilized for both relapse and classification assignments. K nearest neighbors analyzes the name of a particular number of data points circumnavigating a wanted information point to form a forecast around the lesson that information point has a place in. For the fourth model, we have chosen K nearest neighbors Regressor as our machine learning model and have again generated mean absolute error, mean squared error, and root mean squared error metrics.



```
from sklearn.neighbors import KNeighborsRegressor
from sklearn import metrics
#Setup arrays to store training and test accuracies
neighbors = np.arange(1,9)
train_accuracy =np.empty(len(neighbors))
test_accuracy = np.empty(len(neighbors))

classifier = KNeighborsRegressor(n_neighbors=2)
classifier.fit(X_train, Y_train)
```

```
KNeighborsRegressor(algorithm='auto', leaf_size=30, metric='minkowski',
                    metric_params=None, n_jobs=None, n_neighbors=2, p=2,
                    weights='uniform')
```

4.7 Results & Discussion

4.7.1 Accuracy of the model for Aman

Here, we have compared the training and testing accuracy for each of the four algorithms.

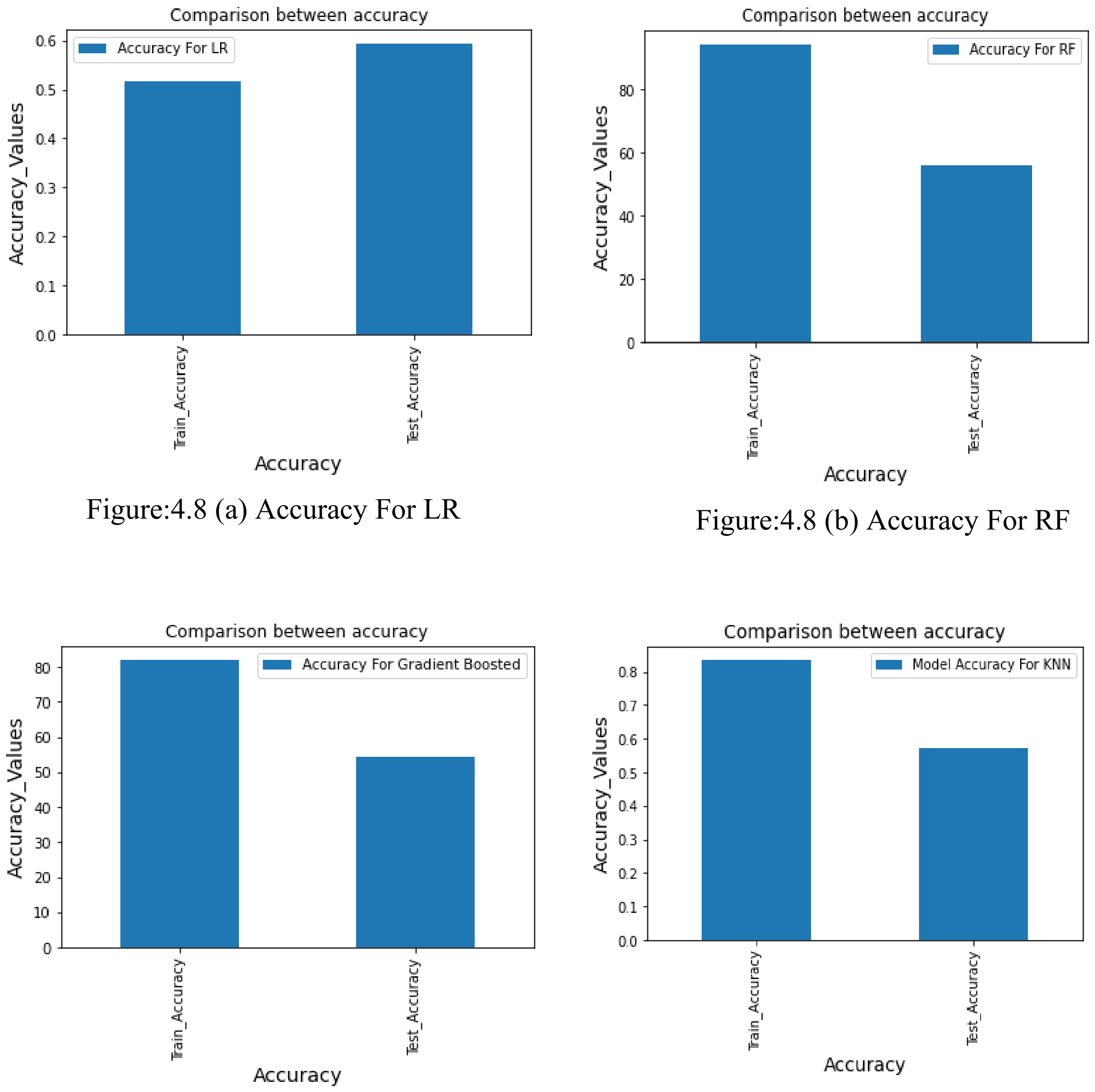


Figure:4.8 (a) Accuracy For LR

Figure:4.8 (b) Accuracy For RF

Figure:4.8 (c) Accuracy For XGB

Figure:4.8 (d)Accuracy For KNN

Figure: 4.8 Accuracy of Different Algorithms for Aman

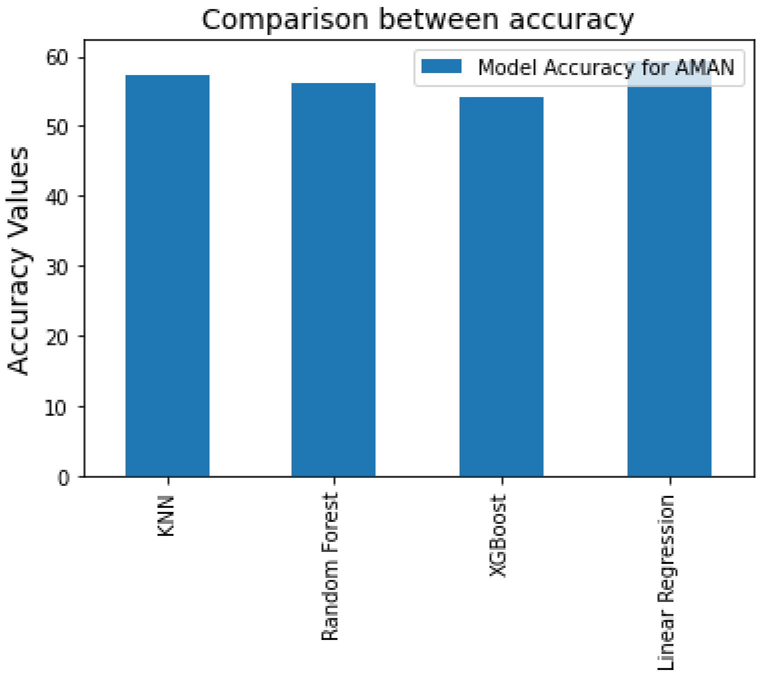


Figure: 4.9 Comparison Accuracy Value for Aman

We have plotted the Accuracy vs Algorithm graph for Aman in Figure: 4.9 to differentiate the model accuracy of different algorithms. From the above picture, we can see that the accuracy of the Linear Regression model is higher than other algorithms which are 61%.

4.7.2 Errors of Aman

Here we have compared the MSE, RMSE and MAE values for four different algorithms.

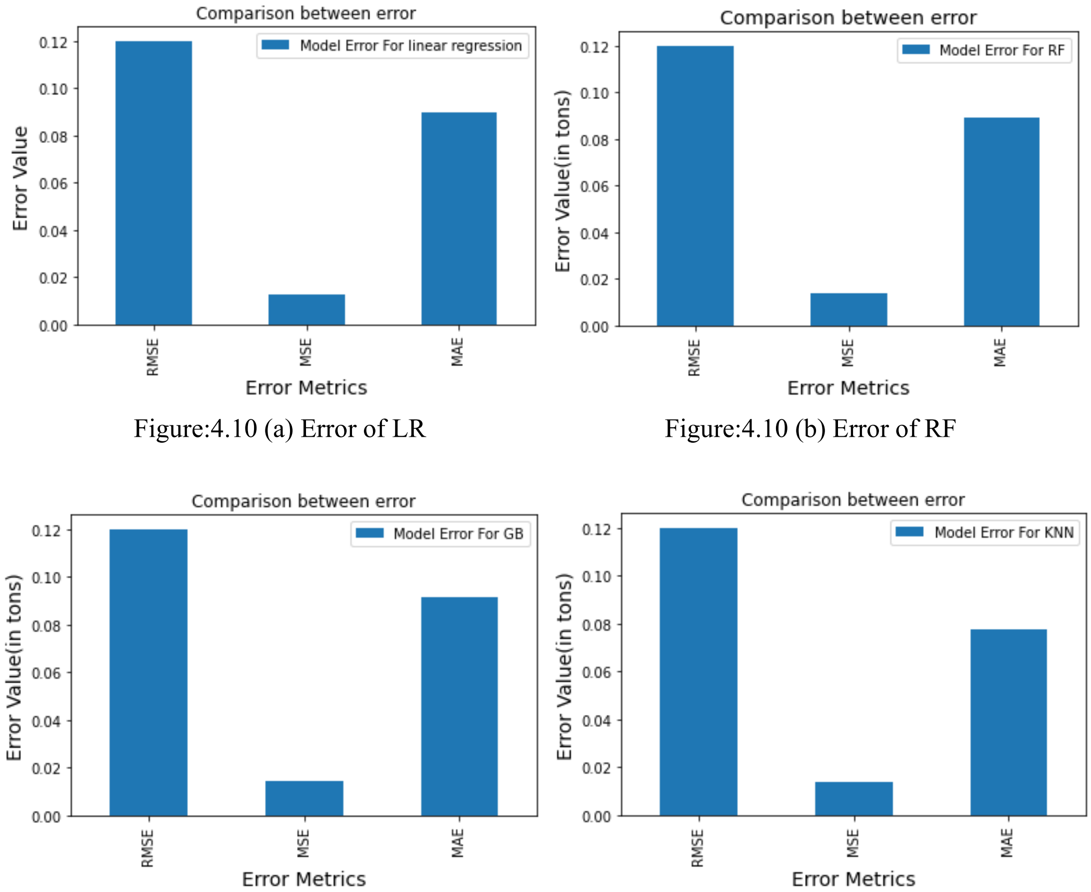


Figure:4.10 (a) Error of LR

Figure:4.10 (b) Error of RF

Figure:4.10 (c) Error of Gradient Boosted

Figure:4.10 (d)Error of KNN

Figure: 4.10 Error values (in tons) of Different Algorithm for Aman

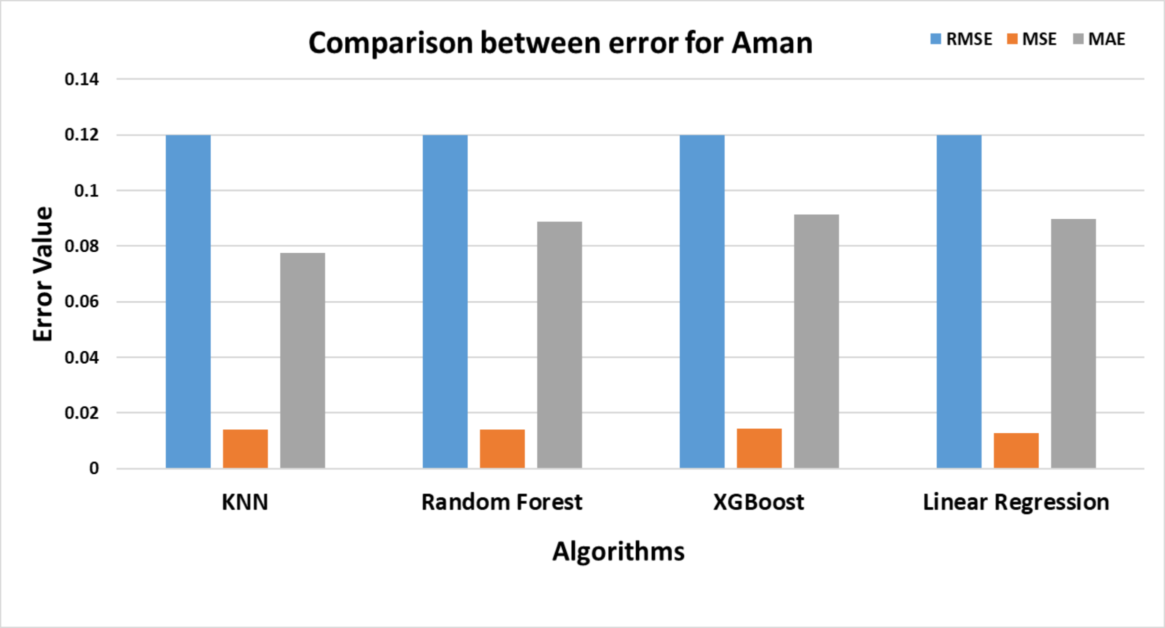


Figure: 4.11 Comparison of Error Values (in tons) for Aman

4.7.3 Scatter plot of Aman

In the scatter plot we kept predicted values in the Y-axis while Actual value was kept in the X-axis. This shows the correlation between both.

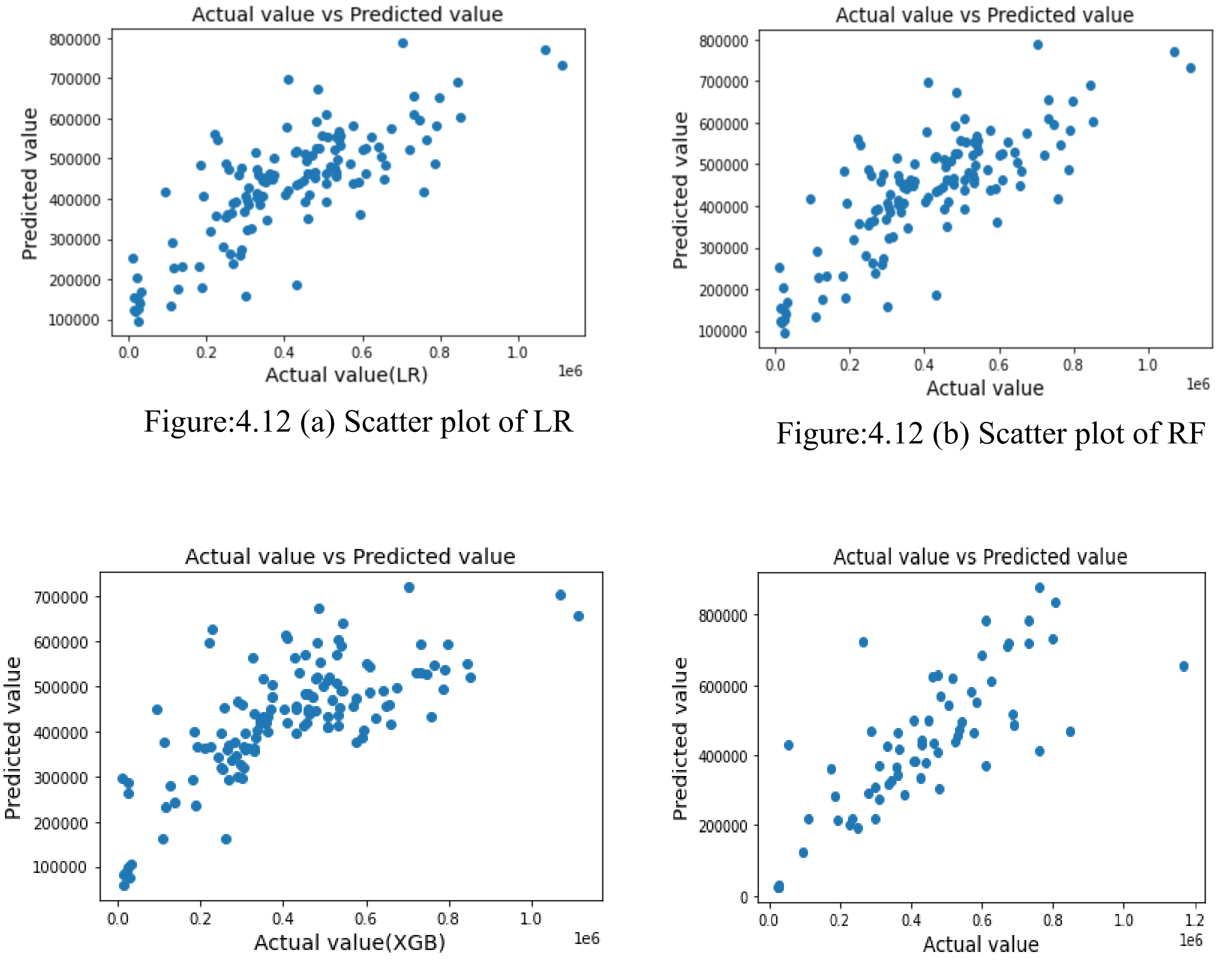


Figure:4.12 (a) Scatter plot of LR

Figure:4.12 (b) Scatter plot of RF

Figure:4.12 (c) Scatter plot of XGB

Figure:4.12(d) Scatter plot of KNN

Figure: 4.12 Scatter of Different Algorithms for Aman

From the above graphs, we can observe that all the predicted and actual values are very close to each other and the regression line and it is also visible that we have a very large dataset for this part of our research.

4.7.4 Accuracy of the model for Aus

Here, we have compared the training and testing accuracy for each of the four algorithms.

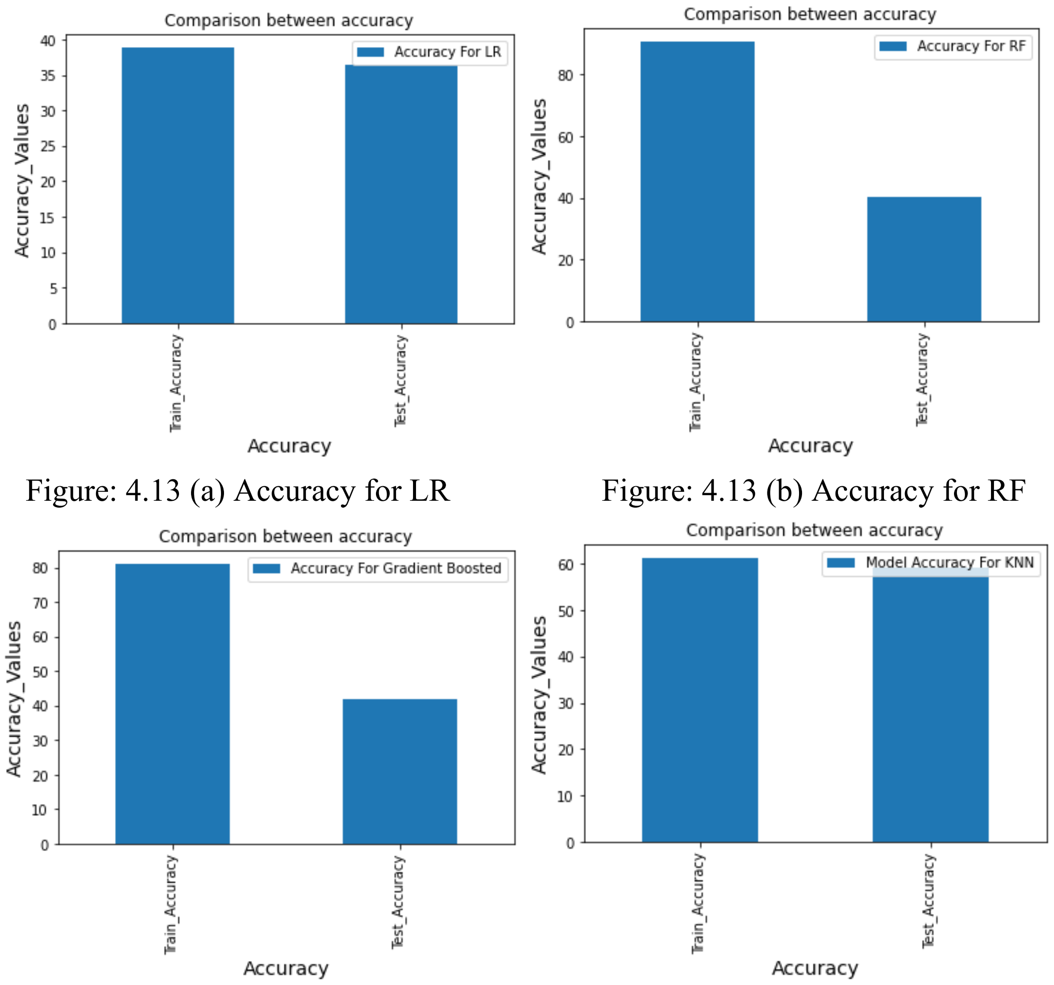


Figure: 4.13 (a) Accuracy for LR

Figure: 4.13 (b) Accuracy for RF

Figure: 4.13 (c) Accuracy for Gradient Boosted

Figure: 4.13 (d) Accuracy for KNN

Figure: 4.13 Accuracy of Different Algorithms for Aus

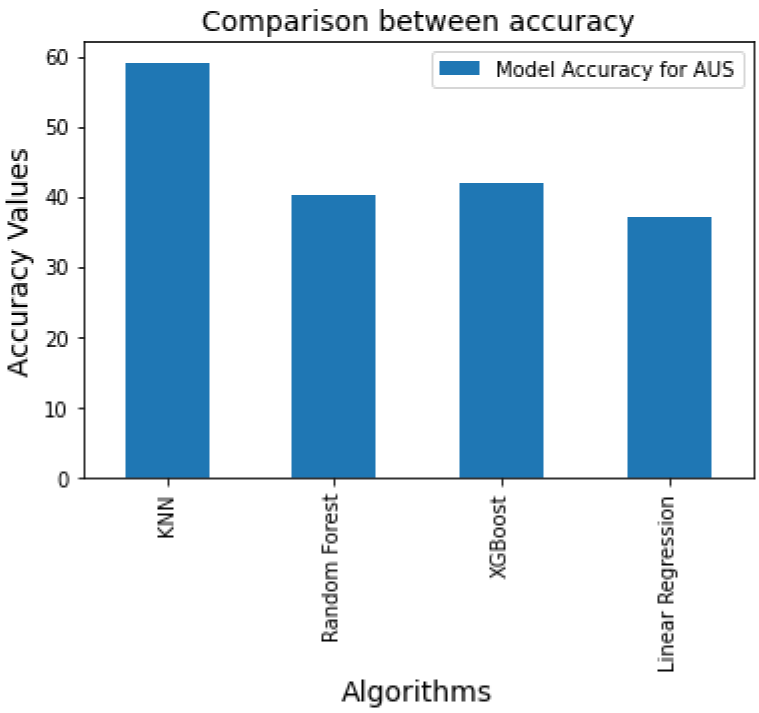


Figure: 4.14 Comparison Accuracy for Aus

We have plotted the Accuracy vs Algorithm graph for Aus in Figure: 4.14 to differentiate the model accuracy of different algorithms. From the above picture, we can see that the accuracy of the KNN model is higher than other algorithms which are 59%.

4.7.5 Errors of Aus

Here we have compared the MSE, RMSE, and MAE values for four different algorithms.

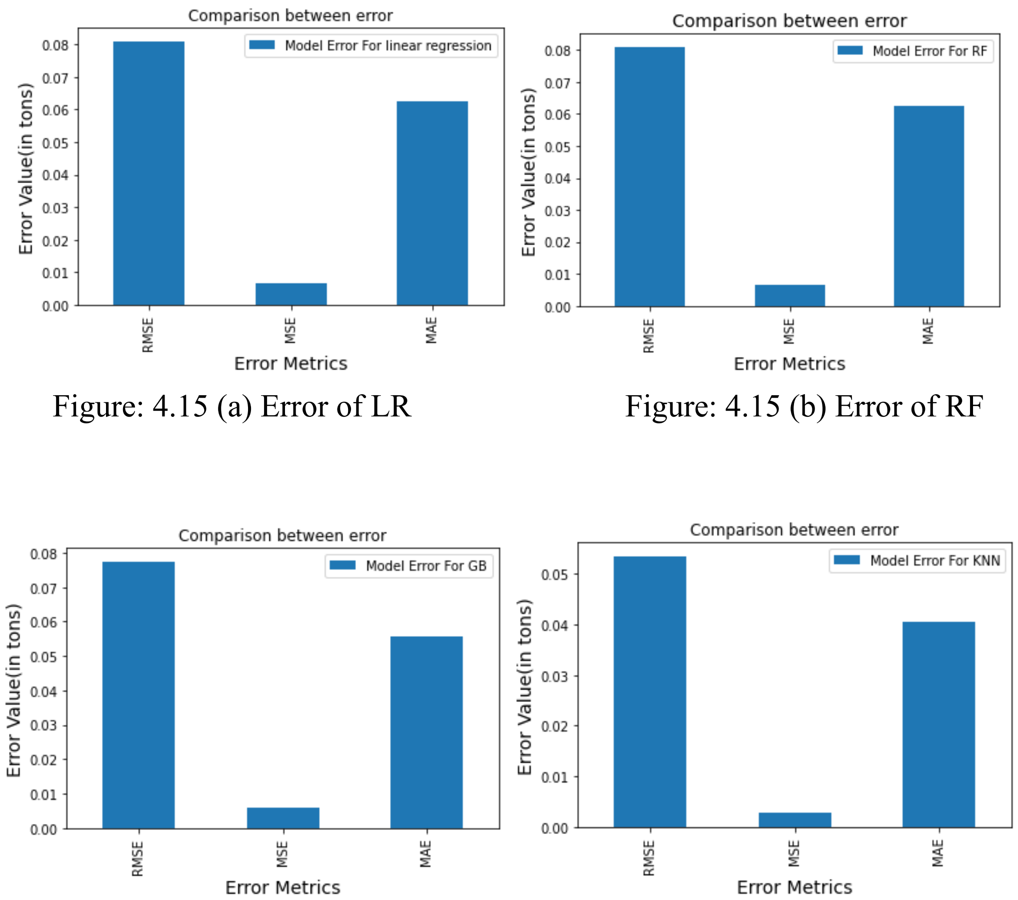


Figure: 4.15 (a) Error of LR

Figure: 4.15 (b) Error of RF

Figure:4.15 (c) Error of XGB

Figure:4.15 (d) Error of KNN

Figure: 4.15 Error values (in tons) for Different Algorithm of Aus

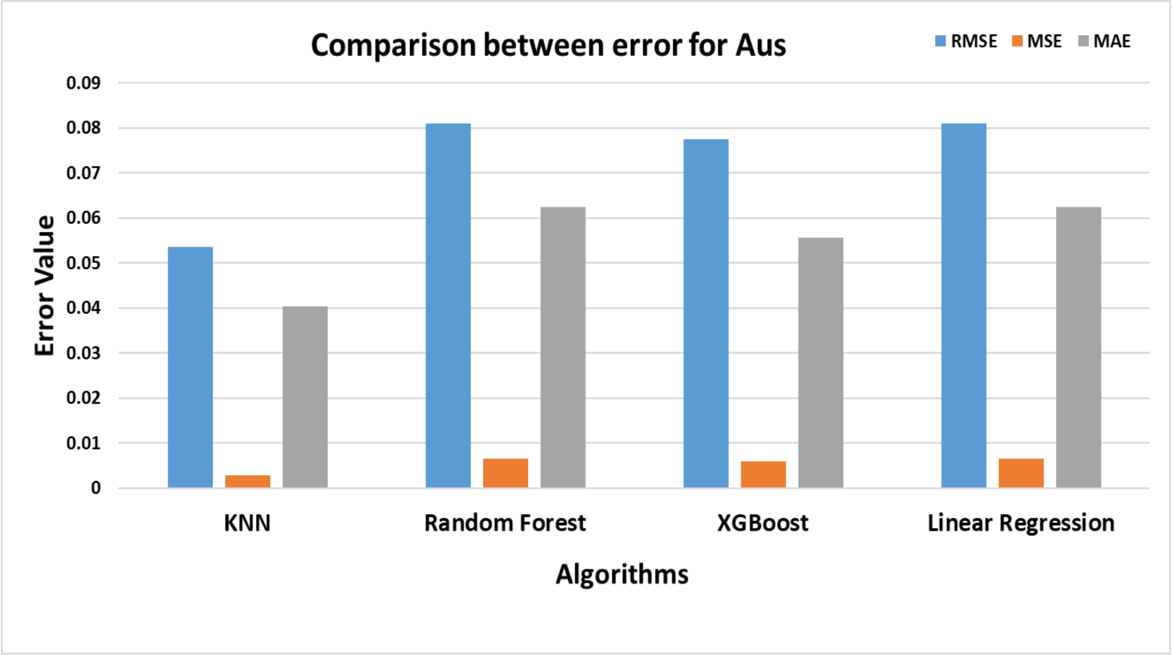


Figure: 4.16 Comparison of Error Values (in tons) for Aus

4.7.6 Scatter plot of Aus

In the scatter plot we kept predicted values in the Y-axis while the Actual value was kept in the X-axis. This shows the correlation between both.

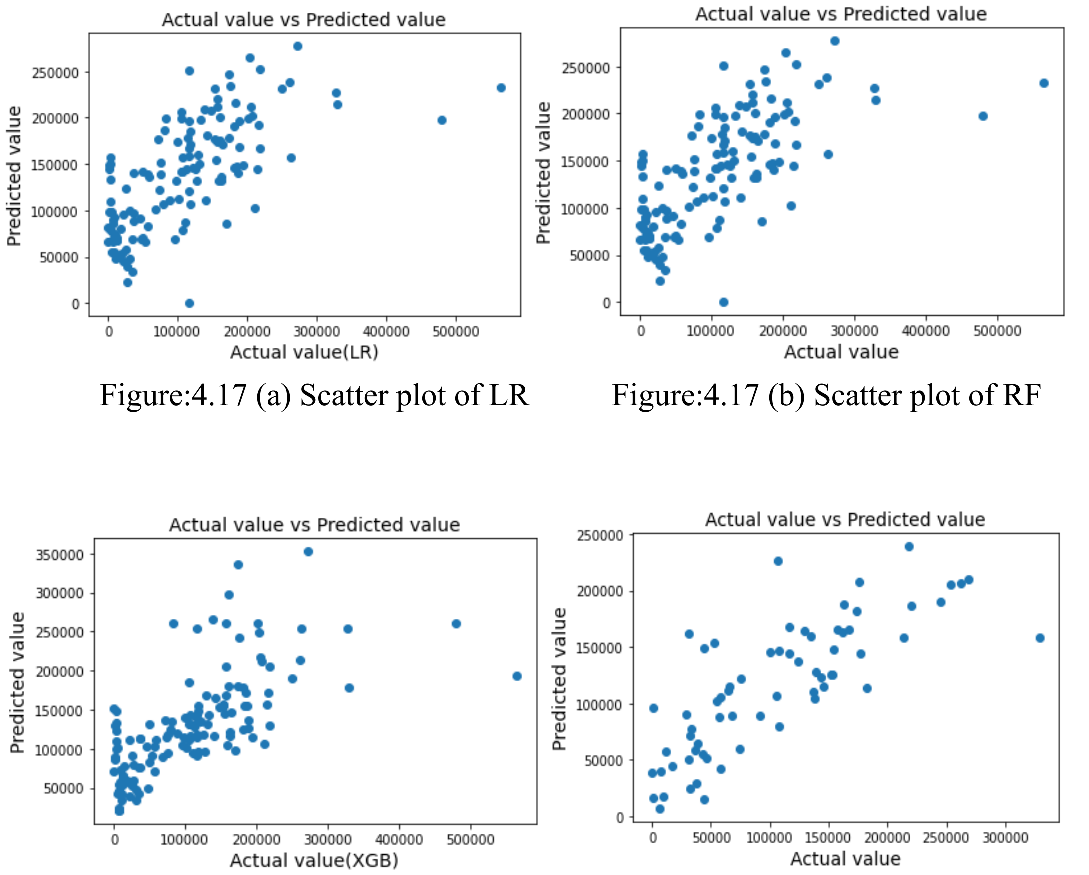


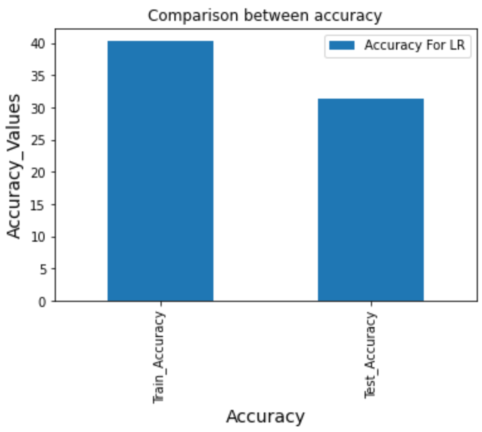
Figure:4.17 (a) Scatter plot of LR Figure:4.17 (b) Scatter plot of RF

Figure:4.17 (c) Scatter plot of XGB Figure:4.17 (d) Scatter plot of KNN

Figure: 4.17 Scatter of Different Algorithms for Aus

4.7.7 Accuracy of the model for Boro

Here, we have compared the training and testing accuracy for each of the four algorithms.



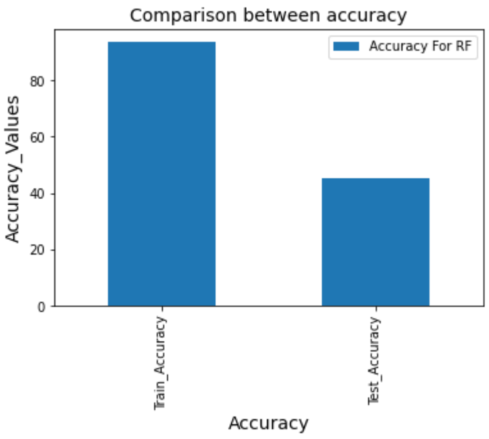


Figure:4.18 (a) Accuracy for LR Figure:4.18 (b) Accuracy for RF

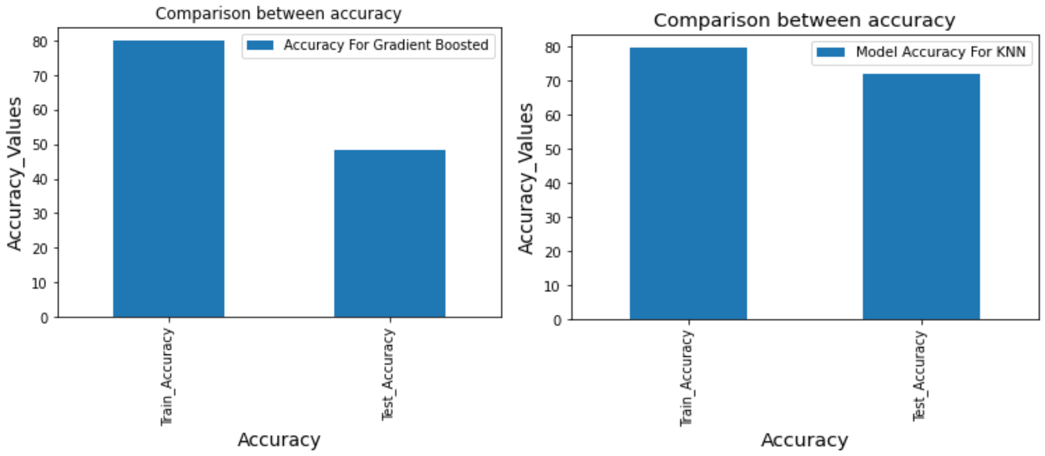


Figure:4.18 (c) Accuracy for XGB Figure:4.18 (d)Accuracy for KNN

Figure: 4.18 Accuracy of Different Algorithms for Boro

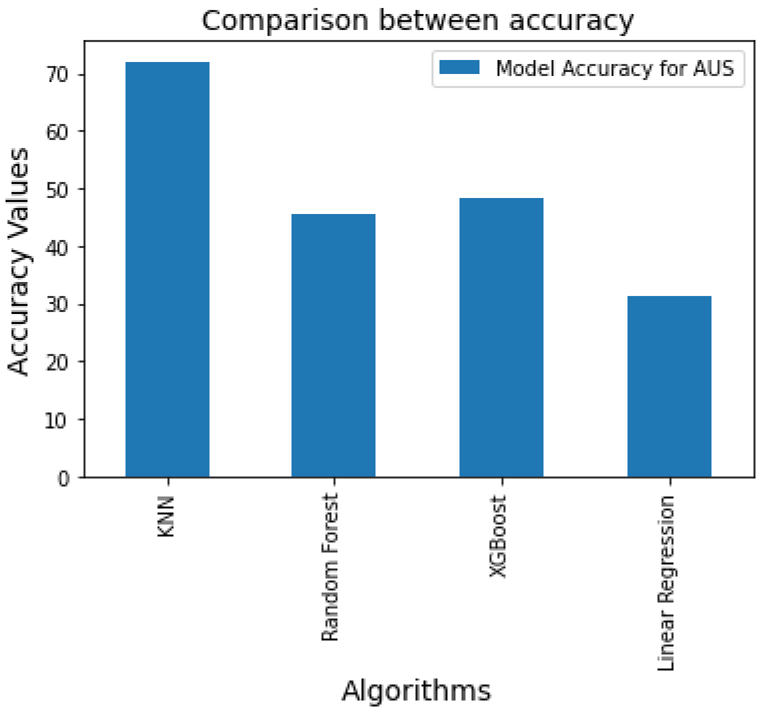


Figure: 4.19 Comparison Accuracy for Boro

We have plotted the Accuracy vs Algorithm graph for Boro in Figure: 4.19to differentiate the model accuracy of different algorithms. From the above picture, we can see that the accuracy of the KNN model, which is 73%, is higher than other algorithms.

4.7.8 Errors of Boro

Here we have compared the MSE, RMSE, and MAE values for four different algorithms.

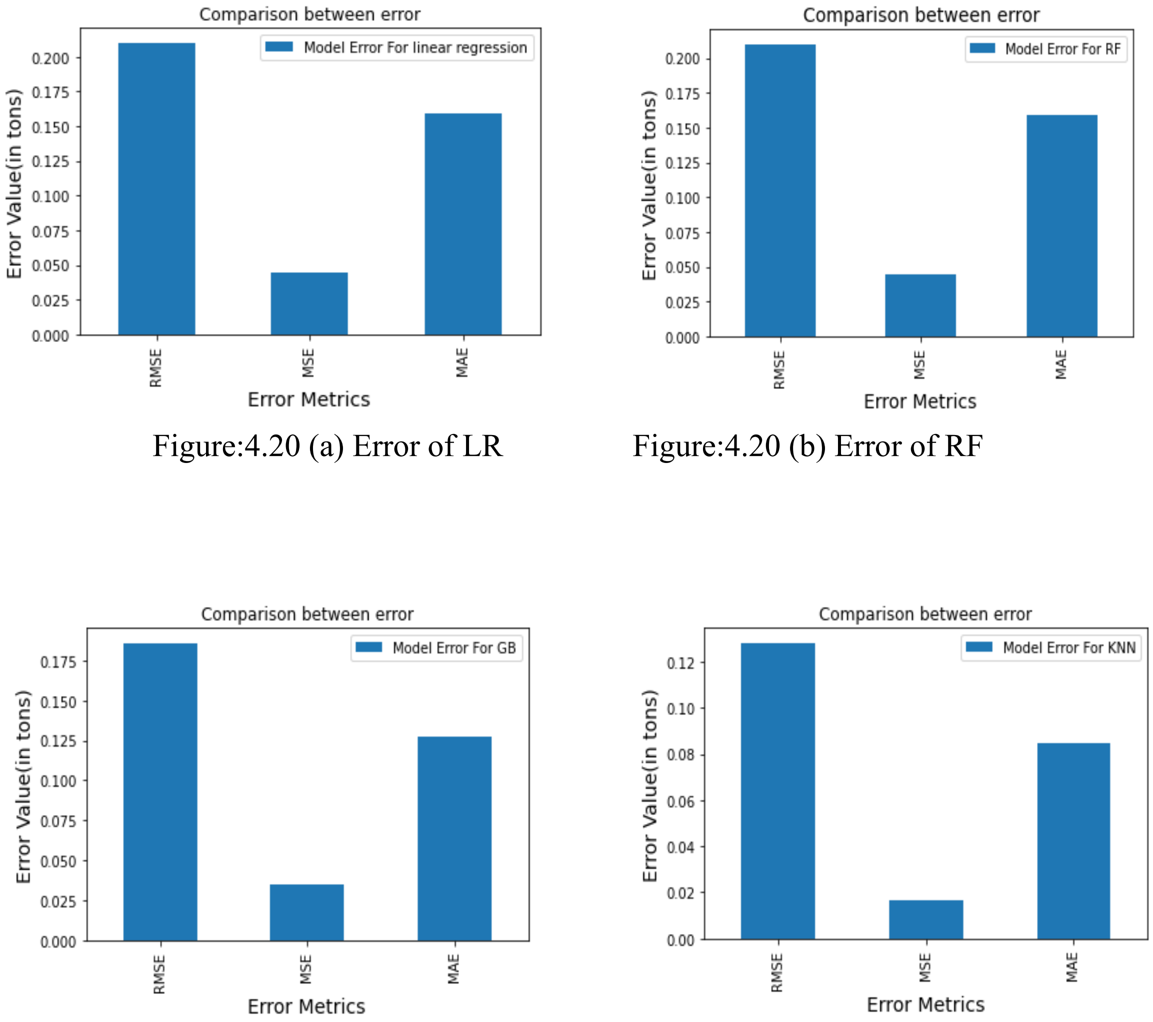


Figure:4.20 (a) Error of LR

Figure:4.20 (b) Error of RF

Figure:4.20 (c) Error of XGB

Figure:4.20 (d) Error of KNN

Figure: 4.20 Error of Different Algorithms for Boro

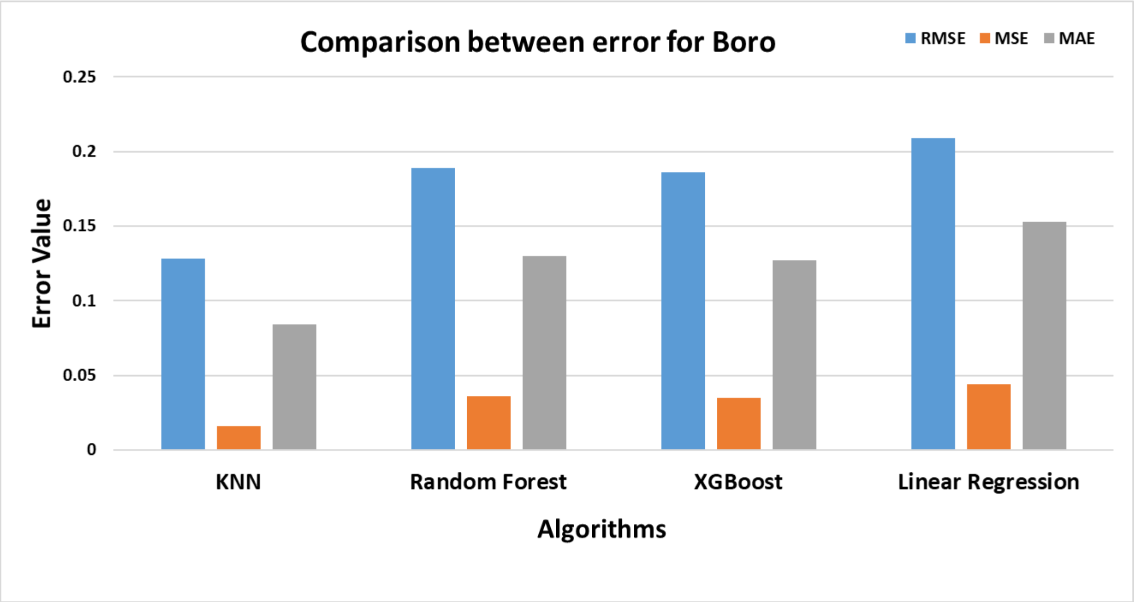


Figure: 4.21 Comparison Error Value for Boro

4.7.9 Scatter plot of Boro

In the scatter plot we kept predicted values in the Y-axis while the Actual value was kept in the X-axis. This shows the correlation between both.

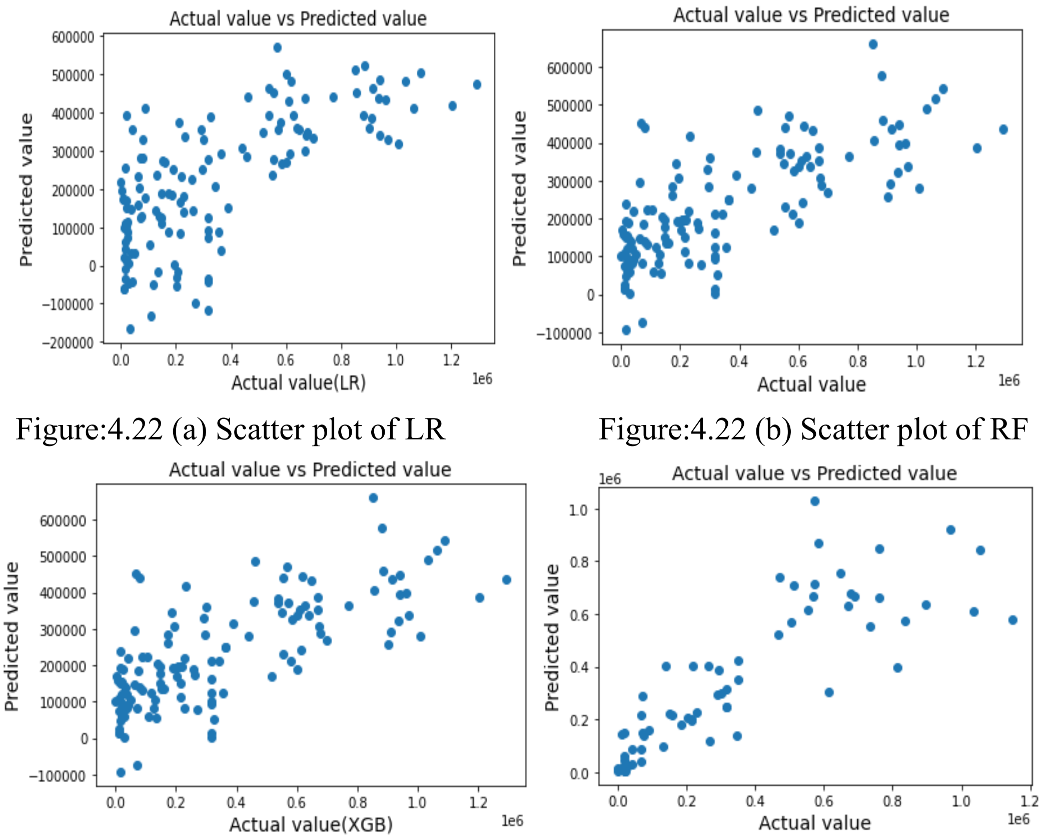


Figure:4.22 (a) Scatter plot of LR

Figure:4.22 (b) Scatter plot of RF

Figure:4.22 (c) Scatter plot of XGB

Figure:4.22 (d) Scatter plot of KNN

Figure: 4.22 Scatter of Different Algorithms for Boro

4.8 Comparison among algorithms

The yield of Aman using the four algorithms such as Linear Regression, KNN, Random Forest, and XGboost yielded the maximum accuracy with Linear regression. KNN generated the least Mean Absolute Error while XGBoost generated the maximum. For Mean Squared Error, Linear Regression generated the least while XGboost generated the most. Linear Regression also generated the least Root Mean Absolute Error while XGboost yielded the most.

Model	Accuracy	Root Mean Squared Error	Mean Squared Error	Mean Absolute Error
Linear Regression	61%	0.111	0.012	0.085
KNN	57%	0.118	0.014	0.078
Random Forest	58%	0.115	0.013	0.084
XGBoost	54%	0.121	0.015	0.091

Table-4.1: Comparison of performance of different models for Aman yield prediction

Model	Accuracy	Root Mean Squared Error	Mean Squared Error	Mean Absolute Error
Linear Regression	36%	0.081	0.007	0.062
KNN	59%	0.054	0.003	0.040
Random Forest	40%	0.079	0.006	0.056
XGBoost	42%	0.078	0.006	0.056

Table-4.2: Comparison of performance of different models for Aus yield prediction

In table-4.2, KNN generated the maximum accuracy for yield prediction while Linear Regression generated the minimum. Random Forest and Linear Regression generated the maximum and similar values of Mean Absolute Error while KNN generated the least. Random Forest and Linear Regression generated the maximum Root Mean Absolute Error while KNN generated the least.

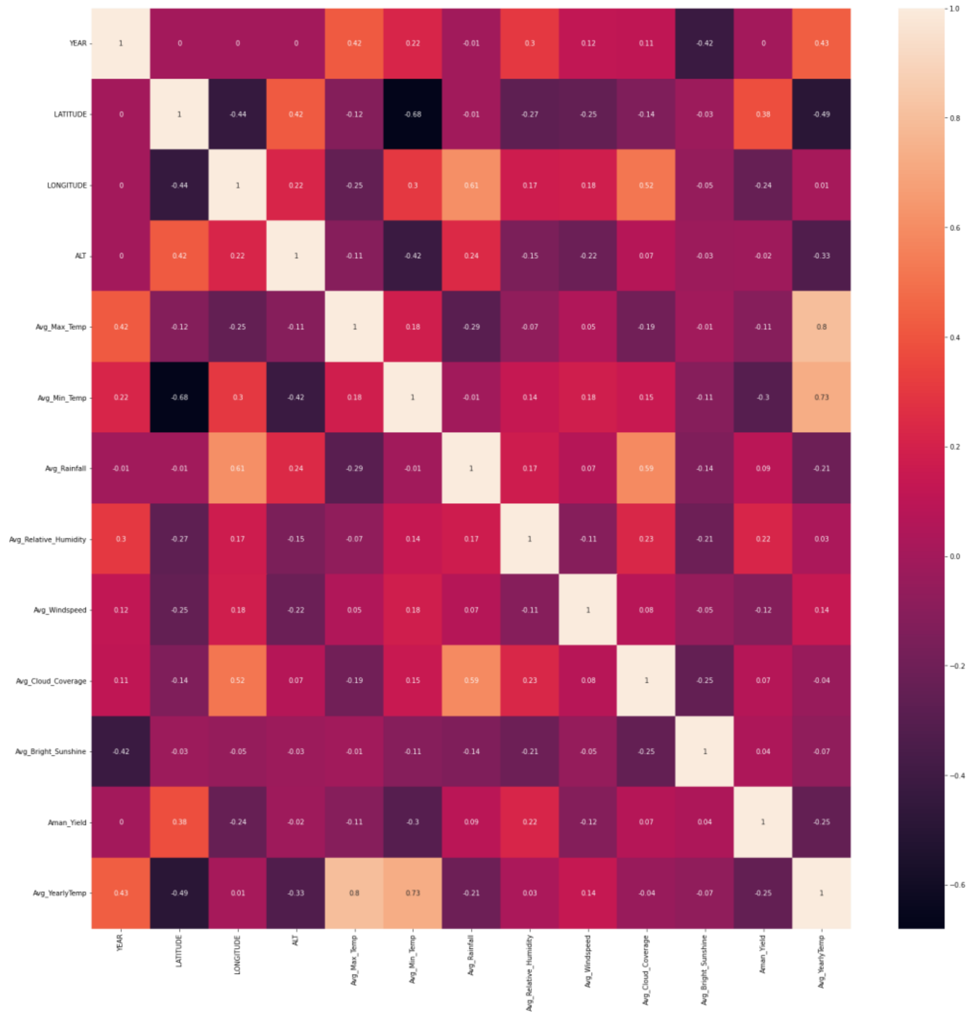
Model	Accuracy	Root Mean Squared Error	Mean Squared Error	Mean Absolute Error
Linear Regression	32%	0.209	0.044	0.153
KNN	73%	0.128	0.016	0.084
Random Forest	45%	0.189	0.036	0.130
XGBoost	48%	0.186	0.035	0.127

Table-4.3 Comparison of performance of different models for Boro yield prediction

In Table-4.3, KNN generated the maximum accuracy while Random Forest generated the least. Random Forest and Linear Regression generated the highest Mean Absolute Error while XGBoost generated the least. Linear Regression generated high Mean Squared Error while KNN generated the least. Linear Regression generated the most Root Mean Absolute Error while KNN generated the least.

From the above tables, we can see that our models have low error rates but only moderate accuracies. We tested Pearson's correlation heatmap to look for a possible cause for this. It can be seen in Figure: 4.3 that our correlation coefficients were not satisfactorily suitable to use in regression algorithms.

Pearson's correlation heatmap shows the coefficient of correlation among the independent multiple variables and with themselves. For proper correlation, the coefficients are required to be either close to 1 or -1. It can be seen that most of the variables have a correlation coefficient below ±0.7, which is the minimum barrier for relevant correlation and hence the models generated less accuracy than optimum.



4.8.5 Accuracy across different algorithms

For Aus, KNN yields the highest accuracy of 59%. For Aman, Linear Regression Yields the maximum accuracy of 61%. And for Boro, KNN yields the highest accuracy of 73%.

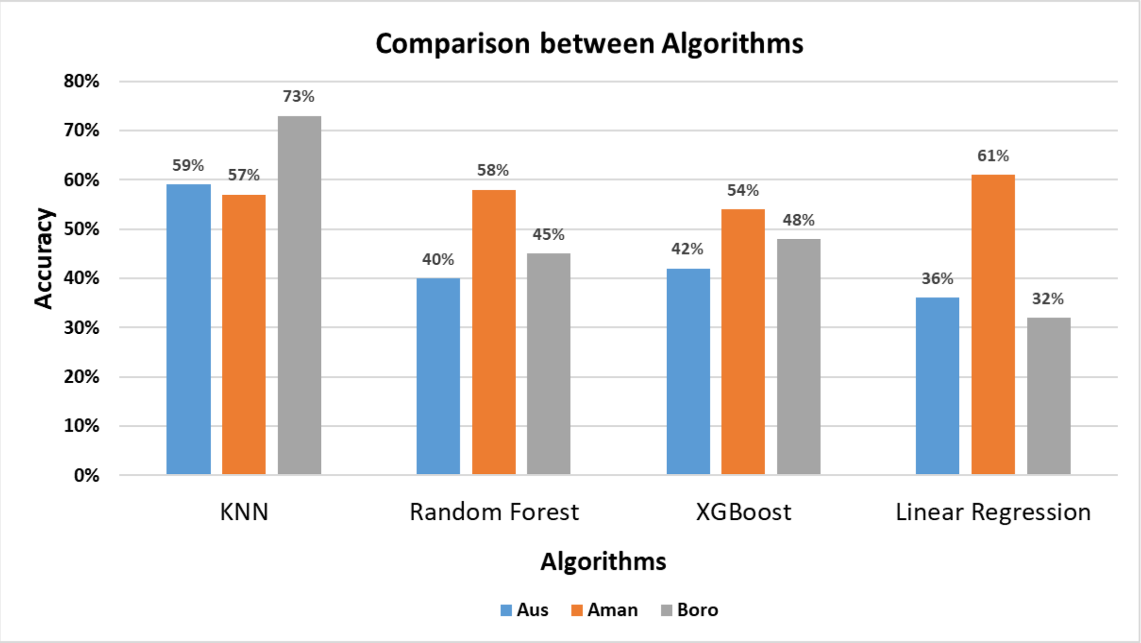


Figure: 4.23 Comparison Between Algorithms

In figure-4.64, we have plotted the graph comparing all the algorithms for Aus, Aman, and Boro where we can see that for Aus, KNN has the highest accuracy among all algorithms which is 59%, and also for the Boro, we have got the best accuracy for the KNN model which is 73%. Moreover, In the case of Aman yield prediction, Linear regression is better than other algorithms with an accuracy of 61%. We have discussed the reason for getting poor accuracy in part 4.8, which is the positive and negative correlation between the weather factors like average maximum temperature, rainfall, humidity with Aus, Aman, and Boro was very low and for that reason, the accuracy of the models is not as expected in this part of our research.

4.9 Comparison with literature

Pyone, H. H. & Swe, T. T (2020) utilized data mining to show the performance of Manhattan based KNN classifier (KNN) to improve the prediction of crop yield under different climatic scenarios by using the Rice crop dataset. They were able to successfully obtain accuracies ranging from 85% to 95%. [48] In our research, we were able to obtain accuracy values of 59% to 73% using KNN and linear regression algorithms for the three types of rice.

Literature			Our Findings		
Model Used	Accuracy	Error	Model Used	Accuracy	Error (RMSE)
KNN [48]	85-95%	N/A*	KNN (Aus)	59%	0.054
			Linear Regression (Aman)	61%	0.111
			KNN (Boro)	73%	0.128

* Not mentioned in literature.

Table 4.4: Comparison between literature and our findings

Due to the lack of correlation between the factors and yield we were unable to obtain high accuracies. Additionally, the data set used was customized by combining data to different sources.

Parameter	Aman	Aus	Boro
LATITUDE	0.38	x	0.32
LONGITUDE	-0.24	x	x
ALT	x	x	x
Avg_Max_Temp	x	x	x
Avg_Min_Temp	x	x	x
Avg_Rainfall	0.09	0.06	x
Avg_Relative_Humidity	0.22	-0.08	0.15
Avg_Windspeed	-0.12	-0.06	-0.01
Avg_Cloud_Coverage	0.07	0.01	0.18
Avg_Bright_Sunshine	0.04	0.13	-0.16
Avg_YearlyTemp	-0.25	-0.41	0.07

**x indicates the features (independent variable that have been dropped)

Table-4.5 Selections of the features matrix of Aman, Aus & Boro yield prediction

It can be noted that the accuracies and errors for an algorithm used are not always proportionate amongst the types of rice crop. For Aman yield prediction the accuracy is 61% and the RSME is 0.111 using linear regression model, for Aus yield prediction the accuracy is 59% and the RSME is 0.054 and for Boro yield prediction the accuracy is 73% and the RSME is 0.128 using KNN for both. We can observe that despite the accuray for Boro being greater than that of Aus and Aman, the RSME is also greater whereas convention dictates that it should be lower. This is due to the independent variable selected for the crops used as can be seen in Table-4.5. For Boro we dropped Avg_Rainfal for the final dataset but used it in the calculations for Aman and Aus and Longitude which we only consider for Aman and not Aus and Boro. These features also had different correlation coefficients to the individual crops.

4.10 Conclusion

To sum up, K-nearest neighbor (KNN) algorithm gives us about 59% and 73% accuracy for Aus and Boro respectively while linear regression shows 61% accuracy for predicting Aman rice yield. The Root Mean Squared Error (RMSE) of Aus and Boro are 0.054 and 0.128 respectively by applying KNN model and for Aman is 0.128 by applying linear regression model. Since there is a lack of correlation between the weather factors and rice crop yield, we were unable to get proper high accuracy, though the error values were comparatively low. Furthermore, the data set of the weather factors is composed of multiple data sets and due to that we could not get desired accuracy for our prediction in this chapter.

Chapter 5
Machine Learning Algorithms in Crop Price Prediction

5.1 Introduction

Rice is the main staple food of Bangladesh, consumed by almost the entirety of its population on a daily basis. Statistics obtained from the Food and Agriculture Organization (FAO) of the United Nations show that in Bangladesh the average annual consumption of rice is 160 to 180 kilograms of rice per person, whereas the average global consumption rate is 50 to 60 kilograms. [49] Therefore, it would be advantageous for us to be able to predict the possible future prices of a product so vital to the day to day lives of so many people. Not only will such predictions into the future retail prices help consumers to know what to expect going into the market and retailers be able to estimate how much to charge but it will be a useful tool for policy makers to make decisions.

5.2 Related Work

ARIMA models have been used to predict prices for years now. Theodoros Koutroumanidis et al. 2009 attempted to predict fuel prices in Greece to help in policy decisions. They obtained a model that had a Mean Absolute Percentage Error of 16.91% and a Root Mean Squared Error of 0.0513. [50]

Banhi Guha and Gautam Bandyopadhyay, 2016 used the ARIMA model to predict the gold prices in India. Using ARIMA (p=1, q=1, d=1) their obtained Root Mean Squared Error was 719.18, Mean Absolute Percentage Error was 3.245% and the Mean Absolute Error was 477.330. [51]

Other notable research papers include Meyler et al., 1998 and their work on forecasting the inflation in Ireland [52]; S. AL Wadi1 et al. 2018 with their utilization of ARIMA model for predicting the stock market closing prices using time series data collected from Amman Stock Exchange (ASE) from January of 2010 till January of 2018 [53]; and Christine Lima and Michael McAleer, 2001 prediction of international travel demand for Australia which is a contrast to the price related forecasting of the other papers. [54]

5.3 Overview of Dataset

We utilized publicly available data for the time series forecasting model. The data sets were obtained from the Food and Agriculture Organization (FAO) of the United Nations. According to their own website, “The Food and Agriculture Organization (FAO) is a specialized agency of the United Nations that leads international efforts to defeat hunger.” [55] Bangladesh has been a member nation of FAO since 12 November 1973. [51] Together they have been working on projects related to agriculture, livestock, fisheries, forestry, food, rural development and climate change.

We chose two similar data sets, one for Dhaka which will be used to fulfil our main aim and one for Delhi for the sake of comparison. Both sets gave us the retail price of rice for the first of the month. We obtained 133 unique values for prices in Dhaka, from January 1st 2008 till January 1st 2019. [56] In contrast, we obtained 229 unique values for the price for rice in Delhi, starting from January 1st 2000 till January 1st 2019. [57] The prices were given in USD which we converted to BDT in our algorithm.

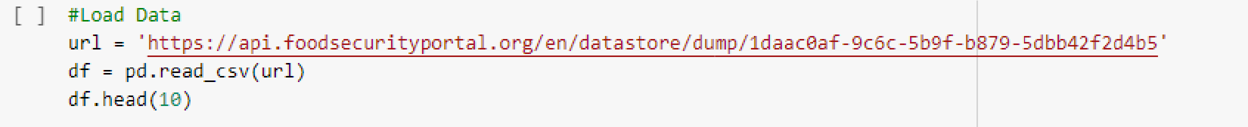
5.4 General Procedure

5.4.1 Importing Library and Modules

To implement the algorithm and demonstrate our results, we have to import the necessary python modules and libraries such as pyplot, numpy, scikit learn which have been previously explained [see Section 3.4.1]. However unique to the ARIMA model is that we have used rcPAram for feature scaling the distribution.

5.4.2 Importing dataset

We import the data from the FAO website by using the link to the data source. We assign the website link as the variable ‘URL’, which we import into the algorithm as a csv file and assign this as a variable df (a shorthand for data frame)



```
[ ]  #Load Data
     url = 'https://api.foodsecurityportal.org/en/datastore/dump/1daac0af-9c6c-5b9f-b879-5dbb42f2d4b5'
     df = pd.read_csv(url)
     df.head(10)
```

5.4.3 Pre-processing the data

Firstly, we have just the drop function to get rid of the data columns that had no relevance to the prediction. Here we are only interested in the date and prices, therefore all other variables are removed.

Following that we check for missing values using the isnull function and displayed the sum total of those values missing from each category using the sum function.

We aggregate the sales by date and index the time series. We multiply the values of the price column with the conversion rate to get the prices in BDT instead of USD.

5.4.4 Displaying data & graph

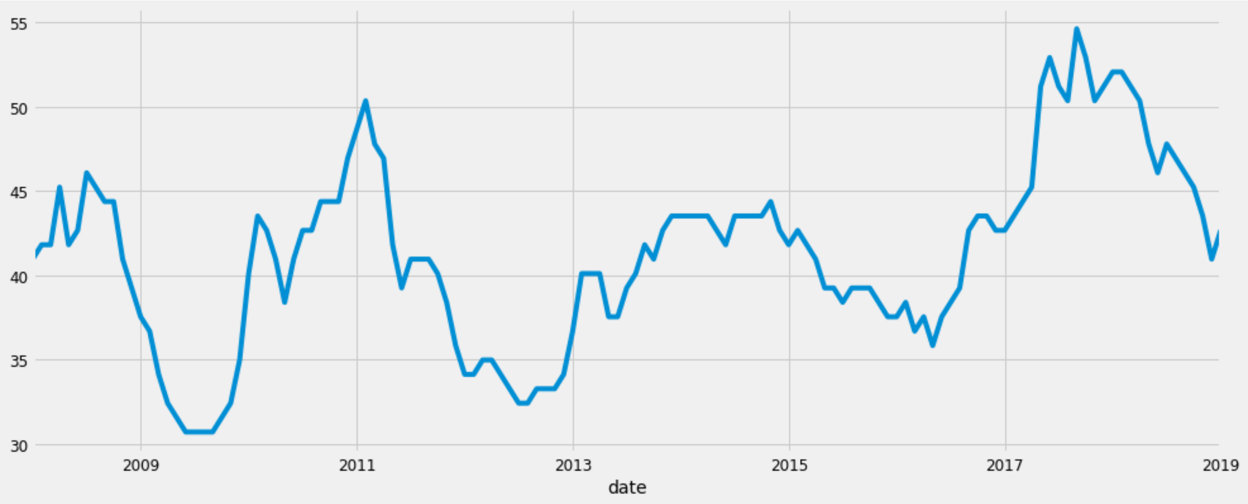


Figure: 5.1 Graph of the data set

When we graphically plot the data in a price (in BDT) as shown against the year as shown in Figure: 5.1. At first glance we notice the time-series does not seem to have a distinguishable seasonality pattern, for example prices decreasing/increasing at the beginning and increasing/decreasing at the end of the year.

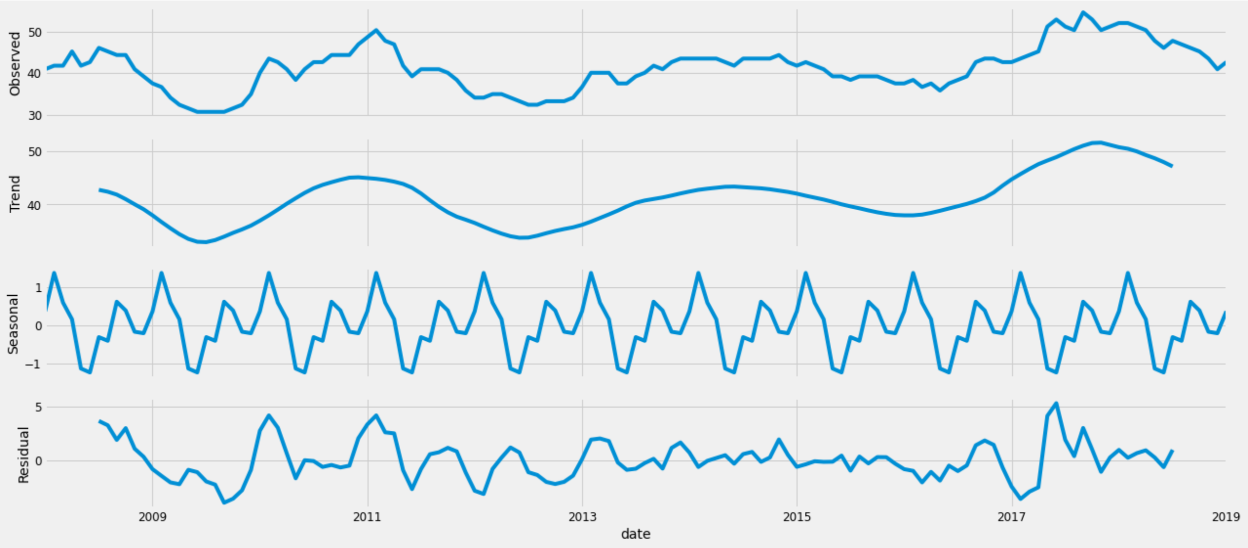


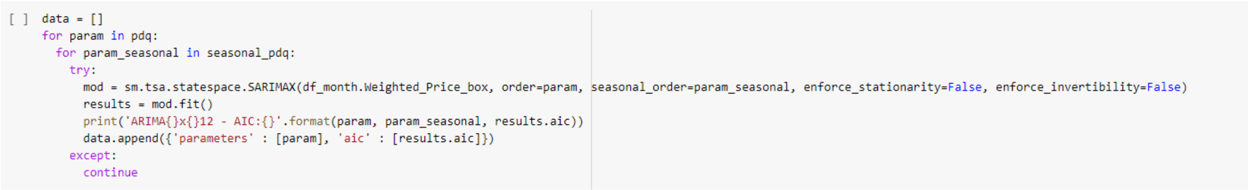
Figure: 5.2 Price change trend, seasonality, and residual noise against time

In Figure: 5.2 we visualize the data further using time-series decomposition that allows us to break down the time series into three components which are trend, seasonality, and residual noise. Here we can confirm there is actually a seasonality in the data.

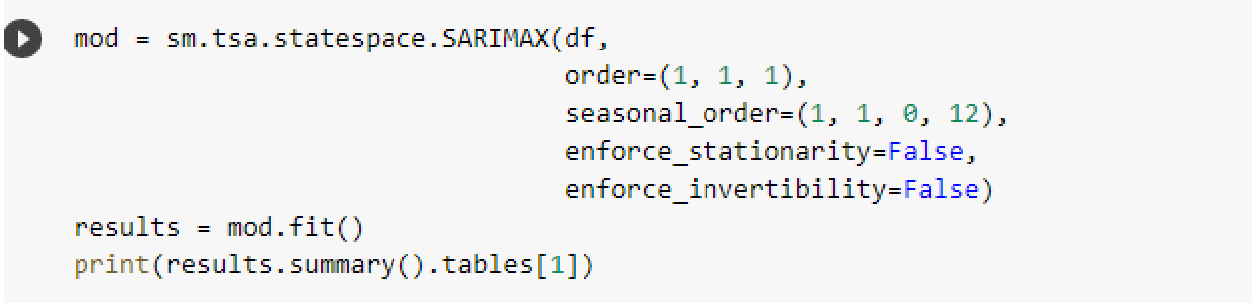
5.4.5 Model Implementation

ARIMA stands for Auto Regressive Integrated Moving Average. Here the auto regressive part deals with a time series display of changing price over a period of time and the moving average is about the overall incremental/decremental change in gradient of the time series analysis. The integration part does not deal with calculus; however, it does deal with the number of step(s) ahead forecast which is the difference between price at (t+n) and price at t to convert the moving average bit from a constant gradient to zero gradient. [9]

The ARIMA model is denoted with the notation ARIMA (p, d, q) where parameters p denotes seasonality, d denotes the trend, and q denotes noise in data. To find the best set of parameters that optimizes the performance of the model, we utilize a grid search, the result of which suggested the use of SARIMAX (1, 1, 1) x (1, 1, 0, 12).



```
[ ]  data = []
     for param in pdq:
       for param_seasonal in seasonal_pdq:
         try:
           mod = sm.tsa.statespace.SARIMAX(df_month.Weighted_Price_box, order=param, seasonal_order=param_seasonal, enforce_stationarity=False, enforce_invertibility=False)
           results = mod.fit()
           print('ARIMA{}x{}12 - AIC:{}'.format(param, param_seasonal, results.aic))
           data.append({'parameters' : [param], 'aic' : [results.aic]})
         except:
           continue
```



```
mod = sm.tsa.statespace.SARIMAX(df,
                                order=(1, 1, 1),
                                seasonal_order=(1, 1, 0, 12),
                                enforce_stationarity=False,
                                enforce_invertibility=False)
results = mod.fit()
print(results.summary().tables[1])
```

The SARIMAX model is an extension of the ARIMAX model which is an abbreviation for Seasonal Autoregressive Integrated Average Exogenous Model. [58] Compared to the ARIMAX model, a SARIMAX model requires 4 additional orders. The first three orders are the same seasonal version of the ARIMA orders except they are denoted by P, Q and D. And the four additional parameters, in our case (1, 1, 0, 12) exert lagged price values from 1 and 1 period and error values from 0 and 12 periods.

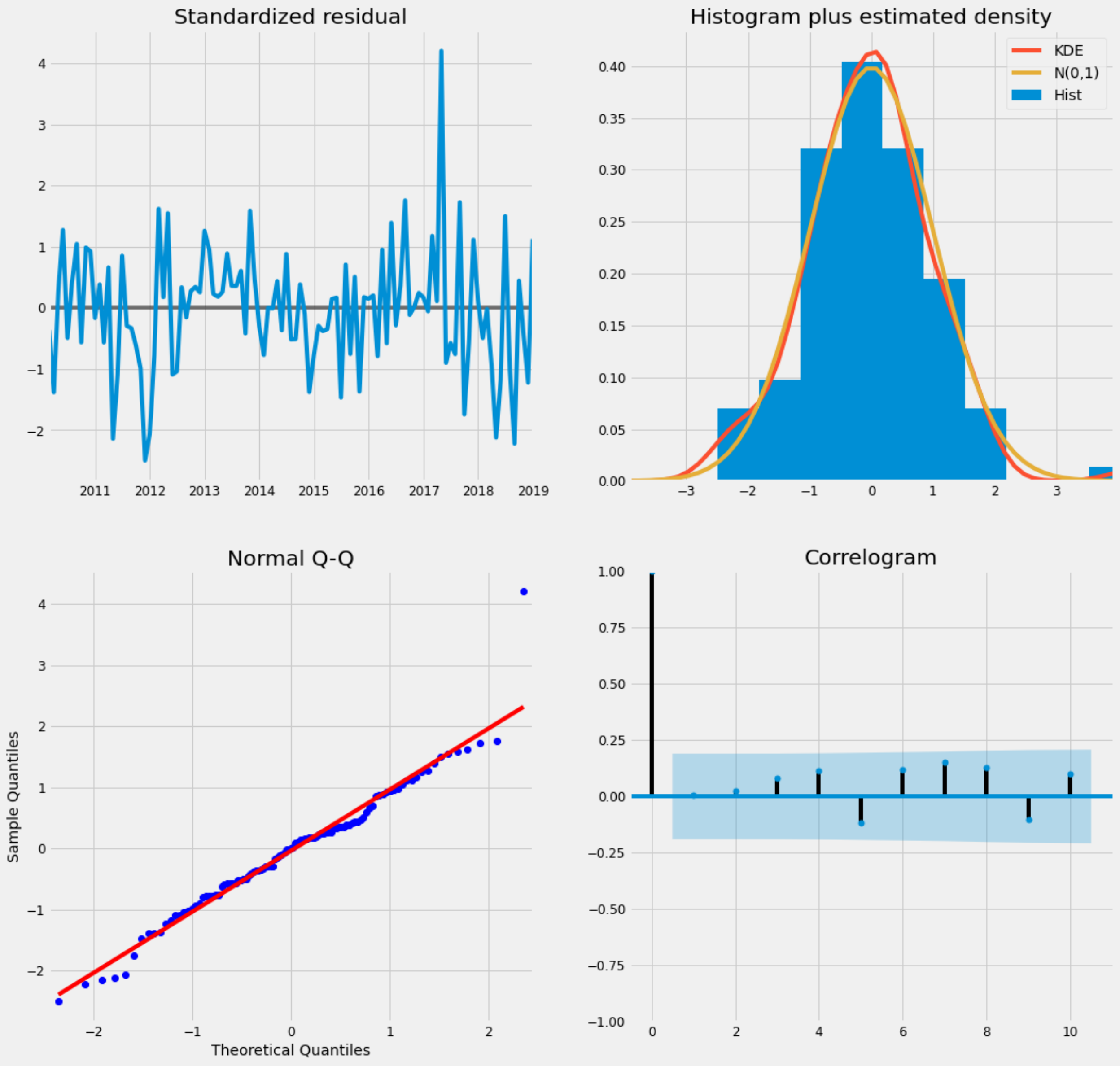


Figure: 5.3 Interpreting the residual plots in ARIMA model

Standardized residual: The error seems to vary around zero and has a uniform variance.
Histogram plus estimated density: It suggests normal distribution with mean zero.
Normal Q-Q: All dots fall around closely to the best fit line and any deviation is an anomaly.
Correlogram: This shows no autocorrelation between residual errors. If any were present it would indicate a model unexplained error pattern.

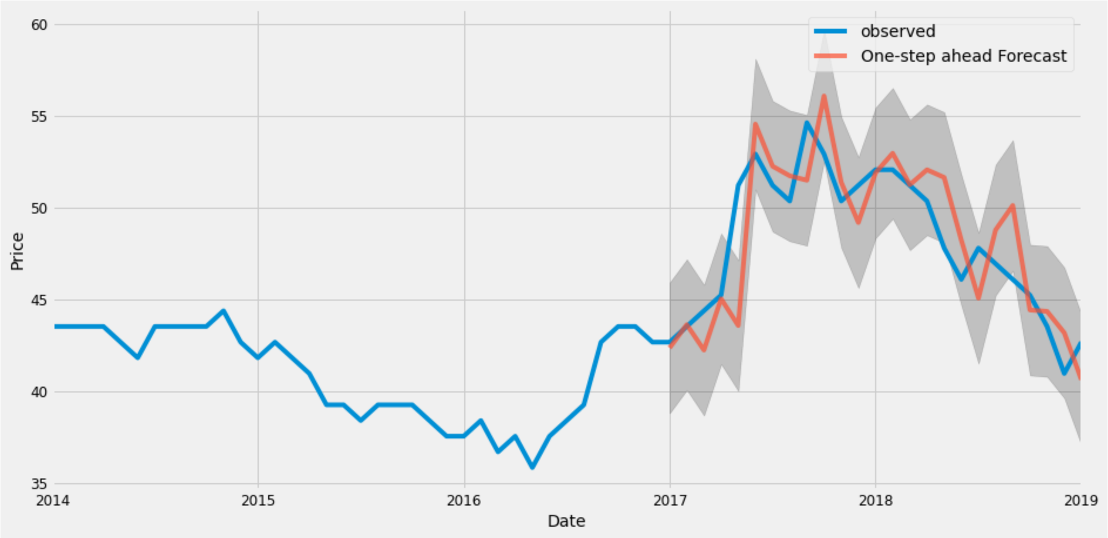


Figure: 5.4 Validations of the data

We validate the accuracy of our forecasts, by comparing the forecasted sales against real sales of the time series, and we forecast the data from 2017–01–01 to the end of the data which is 2019-01-01 as seen in Figure: 5.4

5.5 Result & Discussion

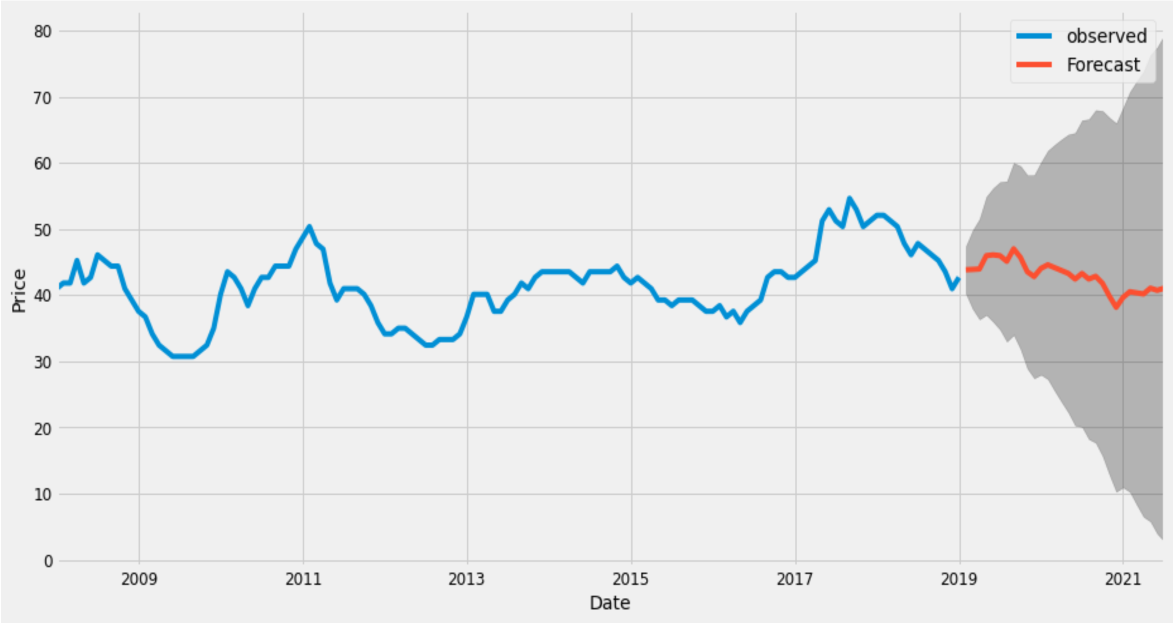


Figure: 5.5(a) Predicted price value for Dhaka

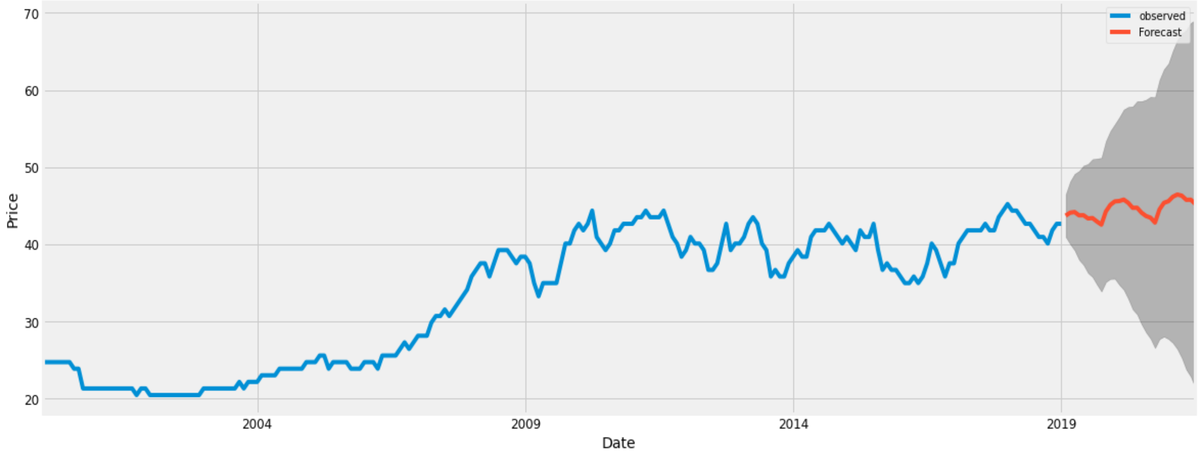


Figure: 5.5(b) Predicted price value for Delhi

In Figure: 5.5(a) The model prediction gives us the possible price of rice in Dhaka for each the first day of month from the 1st of February 2019 till an endpoint that we can set and adjust. However, the greater into the future the end point the less accurate the prediction will be, as seen by the confidence intervals of the model prediction which is depicted by the gray area. It should also be noted that due to the influence of external factors the model cannot account for, the predictions may differ from the real values.

In Figure: 5.5(b) we can see the results for the forecasted model for the rice prices in Delhi. This was mainly used for comparison purposes and the result obtained showed that this model had greater confidence intervals even though it forecasted prices for a greater future date.

5.6 Comparison between results

To evaluate the forecasting capabilities of an ARIMA model MSE, RMSE, MPE and MAPE are most commonly used. The lower these error values, the better is the model at forecasting the future prices.

Region	Mean Squared Error	Root Mean Squared Error	Mean Percentage Error (MPE)	Mean Absolute Percentage Error
Dhaka	6.16	2.48	-0.43%	3.88 %
Delhi	2.12	1.46	0.22 %	2.69 %

Table 5.1: Comparison between models for Dhaka and Delhi

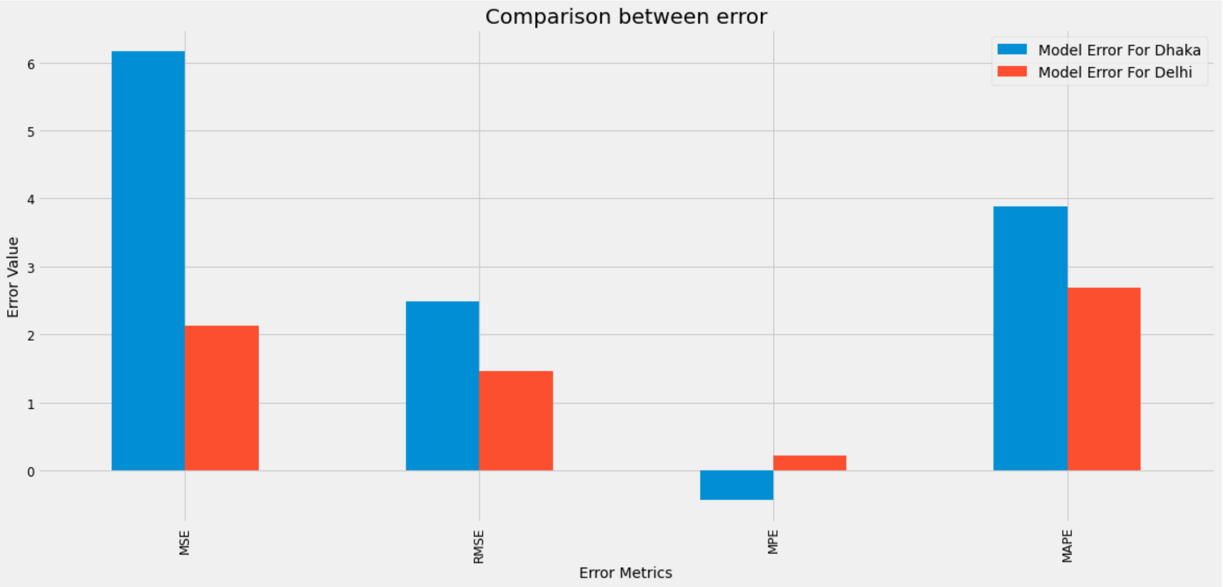


Figure: 5.6 Bar chart comparing the error of our models

As Delhi had more unique data values, the model could interpret the changes in price levels a lot better. There was also a more seasonal fluctuation in price during the recent years as well as in the predicted future values.

Both our models had either better accuracy or less error than the previous models discussed under related works.

5.7 Comparison with Literature

Theodoros Koutroumanidis et al. 2009 [50] had a Mean Absolute Percentage Error of 16.91% compared to our 3.88% for Dhaka and 2.69% for Delhi. However, their Root Mean Squared Error of 0.0513 was significantly lower than our values of 2.48 and 1.46 for Dhaka and Delhi models respectively.

Banhi Guha and Gautam Bandyopadhyay, 2016 [51], had a Root Mean Squared Error value of 719.18, which is much more than that of both our models. The Mean Absolute Percentage Error was 3.245% which was better than our prediction model for Dhaka but not for our model for Delhi.

Literature			Our Findings	
Error Metrics	Theodoros Koutroumanidis et al. 2009 [50]	Banhi Guha and Gautam Bandhyopadhyay, 2016 [51]	Performance Metrics (Error) Dhaka	Performance Metrics (Error) Delhi
Mean Squared Error	N/A*	N/A*	6.16%	2.12%
Root Mean Squared Error	0.0513	7.1918%	2.48%	1.46%
Mean Percentage Error	N/A*	4.7733%	-0.43%	0.22%
Mean Absolute Percentage Error	16.91%	3.245%	3.88%	2.69%

* Not mentioned in literature.

Table 5.2: Comparison between literature and our findings

5.8 Conclusion

In summary we successfully implemented two models that predict the possible price of rice for the first day of every month for the cities of Dhaka and Delhi with a Mean Absolute Percentage Error of 3.88% and 2.69% respectively. Compared with the literature we used for consultation these are fairly good results. The confidence intervals of the model prediction decrease as further into the future we look into as uncertainties increase.

Chapter 6
Conclusion and Future work

6.1 Conclusion

In regards to fertilizer and soil types: we can say that all four algorithms generated fairly accurate analysis of yield based on soil type and fertilizers.

In regards to weather data: all the algorithms generated accuracies which were fairly moderate with acceptable error margins for specific algorithms.

In regards to price: we could successfully implement the machine learning model in spite of the lack of enough price data that is still quite reliable. Overall, further data collection, consolidation and implementation is necessary for better and more satisfactory results, however this is a considerably accurate framework for further more impactful research.

6.2 Future Work and Limitations

6.2.1 In regards to fertilizer and soil types

Since, in our research we have analyzed multiple algorithms to predict the yield of Aus, Aman and Boro using existing old agricultural data, therefore our result may not be that much efficient in the present time. In the near future, this project can be more explored by using real time data which will be collected directly from the land and we can also use a sensor that will be able to sense how fertile a particular soil type is as well as the soil moisture to ensure the maximum crop yield. And by further implementing those machine learning models, it is possible to suggest a suitable land, soil or fertilizer for the maximization of yield.

6.2.2 In regards to weather data

If deployment is implemented, the model can be used to predict the expected yield of a certain year based on the weather data of the year. We can further expand on this by using machine learning algorithms to predict the weather conditions and factors. This way we can predict the possible yield of crops for multiple years in advance. This can assist in the decision making and planning for contingencies for years where the yield might not be satisfactory.

6.2.3 In regards to price

The main hurdle when working on price prediction was that we were unable to get access to large enough data for rice prices or data for any other city but Dhaka. With access to a larger data set from a trustworthy and reliable source, the same model can be used to make future forecasts for the prices much more accurately. If data for other cities and districts can be obtained we can make forecasts for those areas as well and it is a similar case for other commodities.

Reference

[1] A. Géron, "Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd ed)," 2019, Jan 24 2019.

[2] W. McKinney, "Python for Data Analysis (2nd ed)," 2017.

[3] K. Chakraborty and A. A. Hassanien, "Sentiment Analysis on a Set of Movie Reviews Using Deep Learning Techniques," 2019.

[4] S. Acharya, "What are RMSE and MAE?," ed, 2021.

[5] DataTechNotes, "Regression Accuracy Check in Python (MAE, MSE, RMSE, R-Squared)," ed, 2019.

[6] D. A. Bondre and M. S. Mahagaonkar, "Prediction of Crop Yield and Fertilizer Recommendation Using Machine Learning Algorithm," *International Journal of Engineering Applied Sciences and Technology,* vol. 4, no. 5, 2019, September 2019.

[7] H. K. Karthikeya, K. Sudarshan, and D. S. Shetty, "Prediction ofAgricultural Crops using KNN Algorithm," *International Journal of Innovative Science and Research Technology,* vol. 5, no. 5, 2020, May 2020,.

[8] T. S. Yange, C. O. Egbunu, M. A. Rufai, O. Onyekwere, A. A. Abdulrahman, and I. Abdulkadri, "Using Prescriptive Analytics for the Determination of Optimal Crop Yield," *International Journal of Data Science and Analysis,* vol. 6, no. 3, 2020, July 6 2020.

[9] Emeritus. *Introduction to ARIMA: nonseasonal models*. Available: https://people.duke.edu/~rnau/411arim.htm

[10] T. v. Klompenburg, A. Kassahun, and C. Catal, "Crop yield prediction using machine learning: A systematic literature review," vol. 177, ed, 2020.

[11] K. D. Foote, "A Brief History of Machine Learning," in *Smart Data News, Articles, & Education*, ed, 2021.

[12] M. S. S.Dahikar and D. S. V. Rode, "Agricultural Crop Yield Prediction Using Artificial Neural Network Approach," *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering,* vol. 2, no. 1, 2014, January 2014.

[13] E. Manjula and S. Djodiltachoumy, "A Model for Prediction of Crop Yield," *International Journal of Computational Intelligence and Informatics,* vol. 6, no. 4, 2017, March 2017.

[14] S. G. Sangeeta, "Design And Implementation Of Crop Yield Prediction Model In Agriculture," *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH,* vol. 8, no. 1, 2020, January 2020.

[15] X. Chen, Z. Yi, Y. Zhoub, P. Guo, Saeid Gholami Farkoush, and H. Niroomand, "Artificial neural network modeling and optimization of the Solid Oxide Fuel Cell parameters using grey wolf optimizer," *Energy Reports,* vol. 7, pp. 3449-3459, 2021, November 2021.

[16] R. Stureborg, "Easy and Clear Explanation of Neural Nets," in *Artificial Neural Networks for Total Beginners*, ed, 2019.

[17] u. blog, "Neural Network: Architecture, Components & Top Algorithms," ed, 2020.

[18] S. Walczak and N. Cerpa, "Artificial Neural Networks," ed, 2003.

[19] E. Hub, "Artificial Neural Networks (ANN) | Basics, Characteristics, Elements, Types," ed, 2019.

[20] M. Buscema, "STANDARD BACK PROPAGATION," *Back Propagation Neural Networks,* 1998.

[21] S. Jadon, "Introduction to Different Activation Functions for Deep Learning," ed, 2018.

[22] V. Joshi, "Activation Functions," ed, 2019.

[23] "Deep Learning," in *Loss Functions Explained*, ed.

[24] A. Bhandari, "Everything you Should Know about Confusion Matrix for Machine Learning," ed, 2020.

[25] N. Donges, "Gradient Descent: An Introduction to 1 of Machine Learning's Most Popular Algorithms," ed, 2021.

[26] J. Brownlee, "A Gentle Introduction to Mini-Batch Gradient Descent and How to Configure Batch Size," ed, 2017.

[27] Omdena, "Crop Yield Prediction Using Deep Neural Networks and LSTM," 2021, February 28 2021.

[28] S. Ray, "Commonly used Machine Learning Algorithms (with Python and R Codes)," ed, 2017.

[29] C. Hansen, "How to do Linear Regression and Logistic Regression in Machine Learning?," ed, 2019.

[30] Section, "Introduction to Random Forest in Machine Learning," ed, 2020.

[31] J. Brownlee, "How to Create an ARIMA Model for Time Series Forecasting in Python," 2020, December 10 2020.

[32] H. Bansal, "A Quick Way to Learn XGBoost in Machine Learning?," ed, 2019.

[33] F. C. Akyon and E. Kalfaoglu, "Instagram Fake and Automated Account Detection," ed, 2019.

[34] A. Chugh, "MAE, MSE, RMSE, Coefficient of Determination, Adjusted R Squared — Which Metric is Better?," ed, 2020.

[35] S. M, S. Sawant, Vineet.Kishore, and D. S. P, "Crop Production Prediction Using Artificial Neural Network," *Journal of Critical Reviews,* vol. 7, no. 17.

[36] M. L. Carmen, "Retrospective Theses and Dissertations," 1968, Art. no. 3652.

[37] e. a. D.L. Ehret, "Neural network modeling of greenhouse tomato yield, growth and water use from automated crop monitoring data," *Computers and Electronics in Agriculture,* vol. 79, no. 201, pp. 82–89.

[38] B. J. Al, "Artificial neural networks for rice yield prediction in mountainous regions," *Journal of Agricultural Science,* pp. 249–261, 2007.

[39] T. Islam. Agricultural Dataset Bangladesh (44 parameters) [Online]. Available: https://www.kaggle.com/tanhim/agricultural-dataset-bangladesh-44-parameters/metadata

[40] A. Chakrabarty, N. Mansoor, M. I. Uddin, M. H. Al-adaileh, N. Alsharif, and F. W. Alsaade, "Prediction Approaches for Smart Cultivation: A Comparative Study," *Complexity,* vol. 2021, p. 5534379, 2021/04/09 2021.

[41] T. Gupta, "Data Preprocessing in Python," ed, 2019.

[42] J. Brownlee, "Feature Importance and Feature Selection With XGBoost in Python," ed, 2016.

[43] M. A. Hossain, M. N. Uddin, M. A. Hossain, and Y. M. Jang, "Predicting rice yield for Bangladesh by exploiting weather conditions," December 2017 2017.

[44] M. M. Rahman, T. N. Tajnim Jahan, and Z. S. Salma Akter, "Sustainable Rice Production Analysis and Forecasting Rice Yield Based on Weather Circumstances Using Data Mining Techniques for Bangladesh," July 2020 2020.

[45] Y. F. Yahui Guo, X. Z. Fanghua Hao, X. J. Wenxiang Wu, C. R. Bryant, and J. Senthilnath, "Integrated phenology and climate in rice yields prediction using machine learning methods," *Ecological Indicators,* vol. 120, 2021.

[46] R. Bhattacharjee, K. A. Mamun, K. S. Asif, and S. Khan. Weather_Factors_And_Yield_Dataset, [Online]. Available: https://doi.org/10.5061/dryad.qz612jmjp

[47] A. M. u. Kamal. 45 Years Agriculture Statistics of Major Crops (Aus, Aman, Boro, Jute, Potato and Wheat [Online]. Available: http://bbs.portal.gov.bd/sites/default/files/files/bbs.portal.gov.bd/page/16d38ef2_2163_4252_a28b_e65f60dab8a9/45%20years%20Major%20Crops.pdf

[48] H. H. Pyone and T. T. Swe, "Rice crop yield classification by using Manhattan based KNN algorithm," *IJCIRAS,* vol. 2, no. 10, 2020, March 2020.

[49] T. B. Standard, "Per capita rice consumption in Bangladesh to be highest in Asia in 2021: FAO," ed, 2021.

[50] T. Koutroumanidis, K. Ioannou, and G. Arabatzis, "Predicting fuelwood prices in Greece with the use of ARIMA models, artificial neural networks and a hybrid ARIMA–ANN model," *Energy Policy,* vol. 37, no. 9, 2009.

[51] B. Guha and G. Bandyopadhyay, "Gold Price Forecasting Using ARIMA Model," *Journal of Advanced Management Science,* vol. 4, 2016, March 2016.

[52] A. Meyler, G. Kenny, and T. Quinn, "Forecasting irish inflation using ARIMA Models," Research and Publications Department, Central Bank of Ireland, Dublin 22008, Available: https://mpra.ub.uni-muenchen.de/11359/1/MPRA_paper_11359.pdf.

[53] S. AL.Wadi, M. Almasarweh, A. A. Alsaraireh, and e. al, "Predicting Closed Price Time Series Data Using ARIMA Model," *Modern Applied Science,* vol. 12, no. 11, 2018 2018.

[54] C. Lima and M. McAleer, "Time series forecasts of international travel demand for Australia," *Tourism Management,* vol. 23, pp. 389-396, 2001, 6 July 2001.

[55] (1945). *About FAO*. Available: http://www.fao.org/about/en/

[56] F. S. Portal. Bangladesh Rice (Medium) retail monthly prices in the market Dhaka [Online]. Available: https://api.foodsecurityportal.org/en/dataset/bangladesh-rice-medium-retail-monthly-prices-in-the-market-dhaka/resource/1daac0af-9c6c-5b9f-b879-5dbb42f2d4b5

[57] F. S. Portal. India Rice retail monthly prices in the market New Delhi [Online]. Available: https://api.foodsecurityportal.org/en/dataset/india-rice-retail-monthly-prices-in-the-market-new-delhi/resource/b8b5fa2a-3613-5b88-bae0-674357d45707

[58] V. Mehandzhiyski, "What Is a SARIMAX Model?," ed, 2020.

Appendix

A-1	Machine Learning
A-2	Aus
A-3	Aman
A-4	Yield Prediction
A-5	Agriculture
A-7	Artificial Neural Networks (ANN)
A-8	ARIMA
A-9	Linear Regression
A-10	Random Forest
A-11	K-Nearest Neighbor
A-12	XGBoost
A-13	Multidirectional Execution
A-14	Input Layer
A-15	Hidden Layer
A-16	Output Layer
A-17	Weights & Bias
A-18	ReLU
A-19	Gradient descent
A-20	Data Augmentation
A-21	Confusion Matrix
A-22	Recall
A-23	F1 Score
A-24	Mean Absolute Error (MAE)
A-25	Mean Squared Error (MSE)
A-26	Root Mean Squared Error (RMSE)
A-27	Correlation Graph
A-28	Non-Correlation Graph
A-29	Support Vector Machine (SVM)
A-30	backpropagation neural network (BP)