Thesis On

Prediction of Crops Production using Machine Learning Algorithms

Submitted by

Md. Jim Ar Rafi
181400083

A.S.M Imrul Kayes Bhuiyan
181400103

Raihan Munim
181400138

in partial fulfilment of the requirement for the degree of Bachelor of Science in Computer Science and Engineering

Department of Computer Science and Engineering

Faculty of Engineering and Technology

Eastern University

Declaration

We hereby declare that the work is being presented in this project entitled **"Prediction of Crops Production using Machine Learning Algorithms"** in partial fulfilment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering under the faculty of Engineering and Technology, Eastern University, Bangladesh is an authentic record of our own work carried out under the supervision of Mousumi Bala, Assistant Professor, Eastern University**.** It is also declared neither this report nor any part of it has been submitted elsewhere for the award of any kind of degree.

____

Md. Jim Ar Rafi

____

A.S.M Imrul Kayes Bhuiyan

____

Raihan Munim

Approval

The thesis entitled “**Prediction of Crops Production using Machine Learning Algorithms**” submitted by Md. Jim Ar Rafi (181400083), A.S.M Imrul Kayes Bhuiyan (181400103), Raihan Munim (181400138) has been accepted satisfactorily in partial fulfilment of the requirement for the degree of Bachelor of Science Computer Science and Engineering.

Board of Examiners

Mousumi Bala

Assistant Professor

Department of Computer Science and Engineering

Eastern University

Mousumi

Chairman

(Supervisor)

Prof. Dr.Md. Mahfuzur Rahman

Professor and Dean

Faculty of Engineering & Technology

Eastern University

Member

(Ex-Officio)

Muhammad Mahfuz Hasan

Assistant Professor and Chairperson

Department of Computer Science and Engineering

Eastern University

Member

Tabeen Tasneem

Lecturer

Department of Computer Science and Engineering

Eastern University

Member

Abstract

The purpose of this study is to predict crop production using machine learning. As agriculture is an important aspect of Bangladesh's prosperity a large part of the economy depends on it. As we are susceptible to climate change and the ever-growing population, it is important for us to know what can be done for the future of crop production in limited space. Few independent studies have been done to predict production considering factors like harvest area and climate changes. So, our paper focuses on predicting the production of crops depending on temperature, wind speed, humidity, crop type, and harvestable area. We have used various machine learning models and compared the results. Results were compared against five models, Support Vector Machine (SVM), Naive Bayes, Multiple Linear Regression, K-Nearest Neighbor (KNN) and Random Forest. Out of which Random Forest yielded the highest accuracy score.

Table of Content

Content **Page**

List of Figures

Chapter 1

Introduction

1.1 Research Problem

Agriculture was the key development in the rise of sedentary human civilization, whereby farming of domesticated species created food surpluses that enabled people to live in cities. The history of agriculture began thousands of years ago. Bangladesh economy has been surviving on agriculture from its birth stage. Agriculture is one such domain that contributes only around 12.65% to the GDP but has a considerable amount of impact on the Bangladesh economy. As the population of Bangladesh is growing faster, it is important to produce much more crops as it is possible so that we do not need to depend on other countries.

1.2 Research Objective

Conventional agricultural practices and techniques are posing a lot of issues in terms of efficiency, cost-effectiveness and resource utilization. Various kinds of technologies have been applied for the betterment of this area from the very beginning. But sometimes this technology has been used in the wrong way for the lack of proper information and because of that, we could not find perfect results. We are living in the era of modern science. So, it is necessary to use the most advanced technology for cultivation. Bangladesh has the most appropriate weather and fertile land for cultivation. But as there is a huge lack of knowledge, farmers cannot properly minimize their losses. Using the technology, it is possible to decrease the losses if we are able to make a good prediction of our production according to harvested land.

1.3 Scope and Limitations

Over the years due to globalization, agriculture has evolved by adapting the latest technologies and techniques for a better standard of living. Among the technologies and techniques, precision agriculture is one budding technology in the field of agriculture. Machine learning is an appropriate way to make a good prediction. In our thesis, we try to help farmers by our predicted crop production result in the coming years. So as to perform accurate prediction and stand on the inconsistent trends various machine learning classifiers like, Naïve Bayes, Random Forest, Support Vector machine etc. are applied to urge a pattern. We have tried to get the best prediction score from these algorithms considering previous year's information such as harvested area, yields, temperature, humidity and wind speed. We will explain how we use machine learning algorithms with our dataset in further chapters.

Chapter 2

Literature Review

Anakha, Aparna, Jinsu, Rima, has applied machine learning algorithms [1] on the dataset composed of data from different official government websites. It had features like crop name, area, production, temperature, rainfall, humidity and wind speeds. The accuracy of the models that they used is Logistic regression with 87.8%, Naive Bayes with 91.50% and Random Forest with 92.81%. Their goal was to predict the yield and the names of crops from the regions of Kerala. They compared their models and then selected Random Forest as their target model. They also proposed a mobile application that predicts the name of the crop as well as calculates its corresponding yield.

Kiran, Suyog, Smit used random forests to predict crop yield [2]. Their target was to get the maximum yield of crops. They say they analyzed and compared all classification methods and then chose Random Forest Classification technique. In their model, they have used a 10-fold cross-validation technique which indicates high accuracy and correlation between the climate and the crop yield and the accuracy of the model is found 87%. They collected data from various cities of Maharashtra state, crop data, climate, region was essential for their research so they gathered data from government sources. Their goal was also to build a web app that will predict the crop yield based on the factors of climate change.

Ankur, Avinash, Ashwani, Chandan worked on identifying a suitable crop, based on soil and the climate, along with plant disease detection [3]. They applied artificial intelligence-based techniques to predict a suitable crop and also detect the disease in plant leaves, at an early stage. They used Logistic Regression (LR) and Support Vector Machine (SVM) on the dataset. Convolution Neural Network (CNN) with ResNet152 architecture is used for the detection of plant disease. Their dataset consisted of 54,306 photos of healthy and diseased crops. They found crop predictions based on LR and SVM have achieved an accuracy of 93% and 97% respectively across a class of 13 crops. CNN based model predicted disease among 38 different categories from 14 specific crops with an accuracy of 96%. Their goal was to propose an approach that will be beneficial to farmers to identify suitable crops and have plant leaf disease under control.

N. Manjunathan, P. Rajesh, E. Thangadurai, A. Suresh proposed to build a Machine Learning model that can accurately predict the rice crop yield prediction [4]. SVM is used to classify whether rice can grow in that area based on the data from soil, temperature, underground water and rainfall. SVM was also used to classify the crop based on the factors of the area, season. And they also implemented a Web Application that enabled their users to interact with the ML model and make their predictions with their given inputs. The dataset is composed of data collected from Government websites. Their research shows that the average temperature, nitrogen in soil and season had the most effect on rice production. Their Machine Learning model which was trained with Support Vector Machine has managed to obtain an accuracy percentage of 96.5%.

Sunil Kumar, Vivek Kumar, R. K. Sharma has used Support Vector Machine (SVM) to forecast the rice yield [5]. They have experimented on conducting one-against-one multi-classification methods, k-fold cross-validation and polynomial kernel function for SVM training. The dataset has been sourced from the Directorate of Economics and Statistics, Ministry of Agriculture, Government of India. They also have implemented neural networks that have significant application in behavior forecasting, they mentioned the pros and cons of using such methods. Their best prediction accuracy for the 4- year relative average increase has been recorded at 75.06% using a 4-fold cross-validation method. Though this was not very high accuracy, they believe that they could have improved it by redefining the training patterns.

The authors from [6] hypothesized analysis of explorative data and thought of designing different types of predictive models. They have used an ensemble framework, which works by incorporating multiple models and it provides a classifier that outperforms each individual classifier. They incorporated Random Forest, Naive Bayes and Linear SVM as the independent base learners used to create the ensemble model. Using this article, we obtain a comparative study of the different algorithms in data analytics. This helped in determining which algorithm is most appropriate to the proposed system.

In aim of this paper [7] by authors were to help farmers know what is profitable in the perspective of vegetable production. They have used Deep Learning, Machine Learning and Visualization to create an app that can use models like LSTM RNN for crop production, and ARIMA for price production. There were various other options in the app. This paper has helped us focus on the factors that can be beneficial for the prediction of crops in a certain area.

Chapter 3

Methodology

3.1 Proposed Method

In this research, we are predicting the production of crops and from the help of this, we will be able to calculate how many crops can be produced from a certain amount of area so that farmers can understand how to increase the production.

We are using some machine learning algorithms like Support vector machine, K-nearest neighbor, naïve Bayes, random forest etc. to get the best prediction score. We are going to use python language to implement those algorithms. We have collected a dataset from various sources and customized it so that we can work on it properly. Data will be split into two parts for testing and training data using Sklearn train_test_split() function so that there will be no under fitting and overfitting issues. We will use some evaluation matrix such as mean absolute error, f1 score, recall, precession etc. to evaluate the score. Then we will be able to get an accurate prediction from our dataset. Each and every step of our works will be explained briefly in the following chapters.

3.2 Flow chart diagram of working procedure

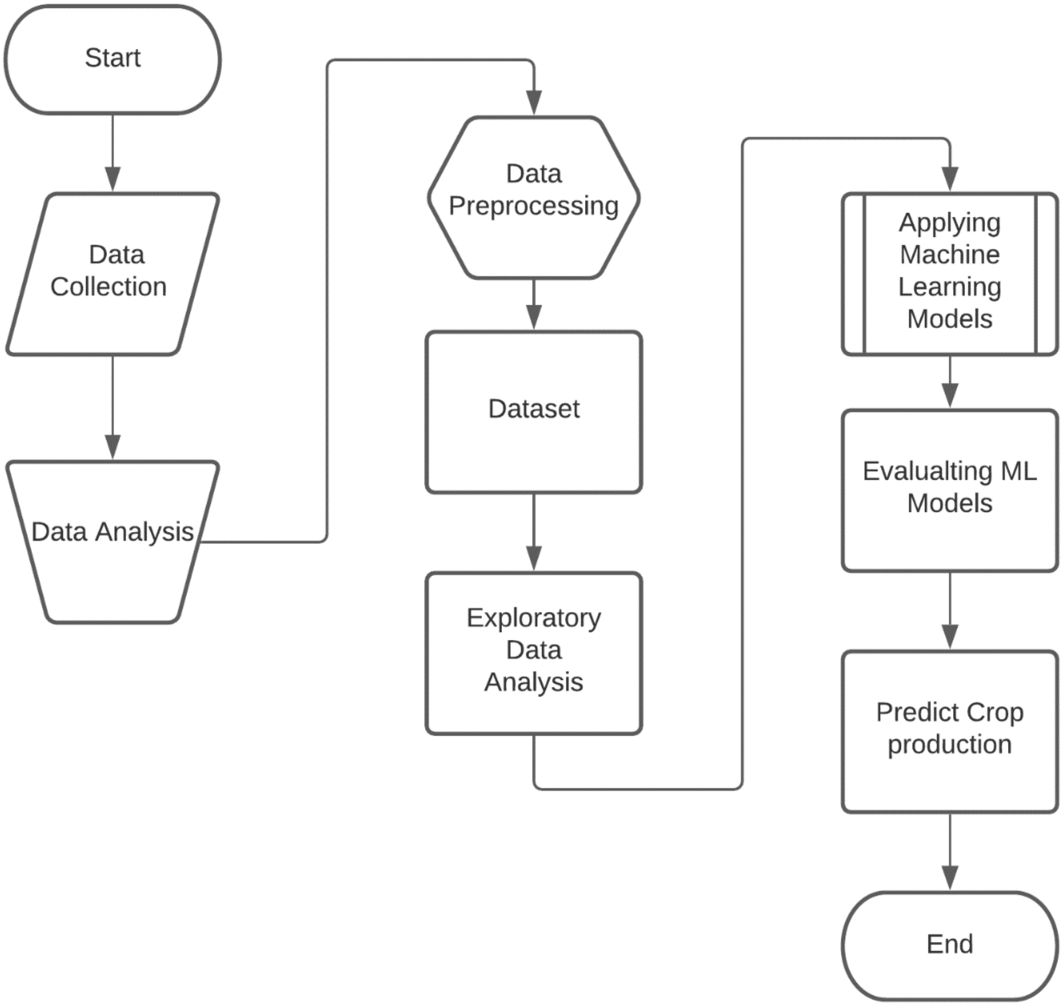


Fig-1 Workflow Diagram

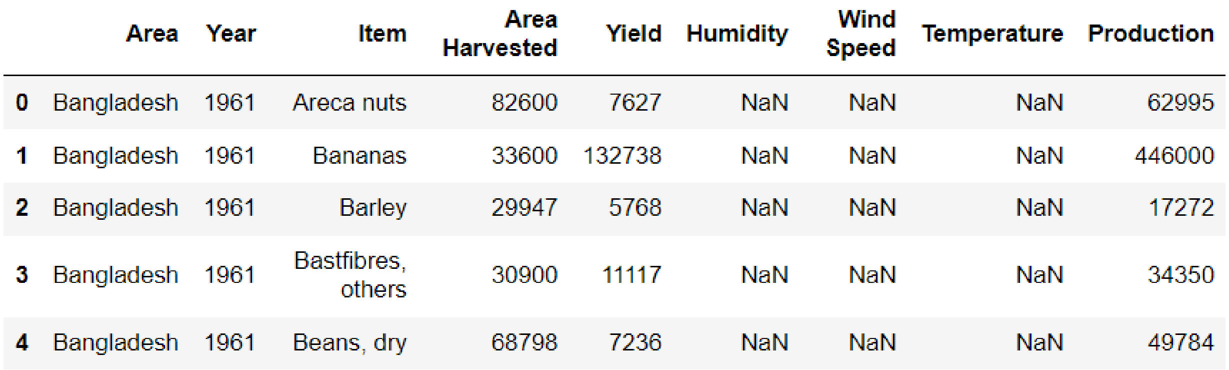
3.3 Data Pre-Processing and analysis

The original dataset can contain lots of missing values so initially, all these should be removed. Missing values are denoted by a *NaN* in the dataset and their presence can deteriorate the value of the entire data and it can reduce the performance.

So, we took the dataset from the Knoema website [8], where it holds agricultural records for almost all the countries that include features like area year-wise harvested, yield, production and their respective unit values. The dataset had flaws like missing entire rows of data, data that are not relevant to our research. We modified the dataset according to our requirements and processed it manually.

Collected and customized the data into a workable dataset that could be applied to a handful of machine learning models. We have used python in Jupyter Notebook for the implementation and testing phase. We imported necessary libraries such as NumPy, Pandas, Matplotlib, Sk-learn etc.

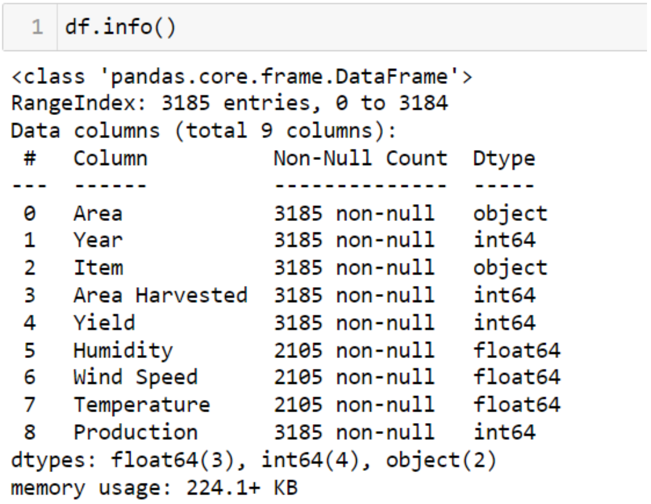
First, we merged all the collected data into a workable format.



	Area	Year	Item	Area Harvested	Yield	Humidity	Wind Speed	Temperature	Production
0	Bangladesh	1961	Areca nuts	82600	7627	NaN	NaN	NaN	62995
1	Bangladesh	1961	Bananas	33600	132738	NaN	NaN	NaN	446000
2	Bangladesh	1961	Barley	29947	5768	NaN	NaN	NaN	17272
3	Bangladesh	1961	Bastfibres, others	30900	11117	NaN	NaN	NaN	34350
4	Bangladesh	1961	Beans, dry	68798	7236	NaN	NaN	NaN	49784

Fig-2 Dataset before preprocessing

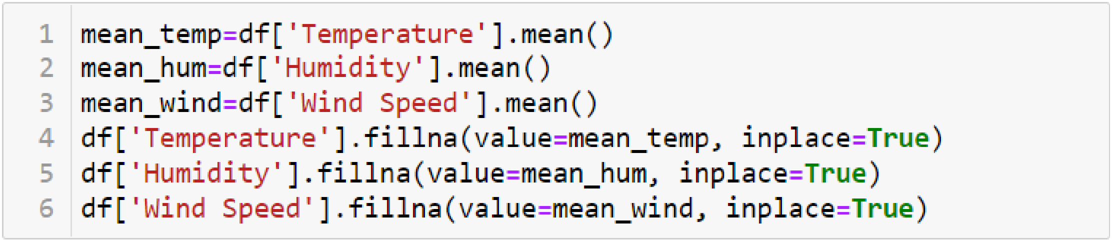
Then we did data analysis on the dataset, looked for faults and errors we can improve upon. Categorical data is of two types, one *nominal*, they don't have an intrinsic order and the other one is *ordinal*, where the data follows an intrinsic order. We filled all the NaN with mean values based on the other years, as the weather side of data lacked before 1981. Following figure shows the data frame information.



```
df.info()
```

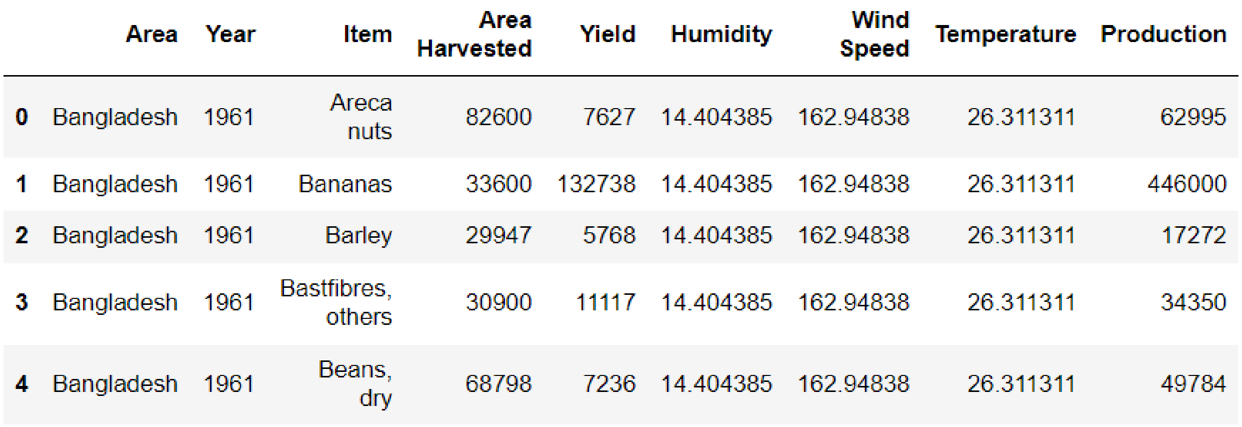
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3185 entries, 0 to 3184
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Area            3185 non-null   object 
 1   Year            3185 non-null   int64  
 2   Item            3185 non-null   object 
 3   Area Harvested  3185 non-null   int64  
 4   Yield           3185 non-null   int64  
 5   Humidity        2105 non-null   float64
 6   Wind Speed      2105 non-null   float64
 7   Temperature     2105 non-null   float64
 8   Production      3185 non-null   int64  
dtypes: float64(3), int64(4), object(2)
memory usage: 224.1+ KB
```

Fig-3: Dataframe Information



```python
mean_temp=df['Temperature'].mean()
mean_hum=df['Humidity'].mean()
mean_wind=df['Wind Speed'].mean()
df['Temperature'].fillna(value=mean_temp, inplace=True)
df['Humidity'].fillna(value=mean_hum, inplace=True)
df['Wind Speed'].fillna(value=mean_wind, inplace=True)
```

Fig-4: Data analyzed and fixed



	Area	Year	Item	Area Harvested	Yield	Humidity	Wind Speed	Temperature	Production
0	Bangladesh	1961	Areca nuts	82600	7627	14.404385	162.94838	26.311311	62995
1	Bangladesh	1961	Bananas	33600	132738	14.404385	162.94838	26.311311	446000
2	Bangladesh	1961	Barley	29947	5768	14.404385	162.94838	26.311311	17272
3	Bangladesh	1961	Bastfibres, others	30900	11117	14.404385	162.94838	26.311311	34350
4	Bangladesh	1961	Beans, dry	68798	7236	14.404385	162.94838	26.311311	49784

Fig-5: After preprocessing of the dataset

With ordinal data, we use an ordinal encoding. But our data required a nominal encoding where we used *One Hot Encoding.*

As per the dataset the solution is, we create a separate column for each category. We convert strings to vectors of each category and that is how One Hot Encoding is done. No matter how many categories there are, we'll have to make those columns.

Another concept surfaces, that is a dummy variable trap, generally when we do one-hot-encoding, we remove one of the columns from the newly created categorical columns.
If there are *n* number of columns, then we will keep the *n-1* number of columns. This brings up concepts like *multicollinearity*, meaning the input columns have a mathematical relationship, if yes, they are dependent on each other and it should be avoided. The columns should be independent, and y is a dependent column as it depends on the independent columns. We cannot let the columns have a mathematical relationship. The newly created columns are called dummy variables which raises the problem of multicollinearity, that is why it is called the Dummy variable trap. The solution to that is to keep *n-1* columns.

We have used One Hot Encoding using Sklearn and removed multicollinearity by removing the categories from the column with drop first and setting the data type as an integer.

After preprocessing we have split our data set into three parts, train set, test set and validation set. Using Sklearn, train_test_split in the ratio of 75% and 25% as training and testing and had a random state of 2, first 8 columns as input and last column as output/target value.

The validation set has been used on a small set of data to check the behavior of the models according to our dataset, then used K-Fold Cross Validation metrics to determine which models would be best for the data at hand.

3.4 Comparison and Selection of Machine Learning Algorithms

Machine Learning is the best technique that gives a better practical solution to crop production problems. There are a lot of machine learning algorithms used for predicting crop production. In this paper we include the following machine learning algorithms for selection and accuracy comparison.

3.4.1 Linear Regression

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering and the number of independent variables being used.

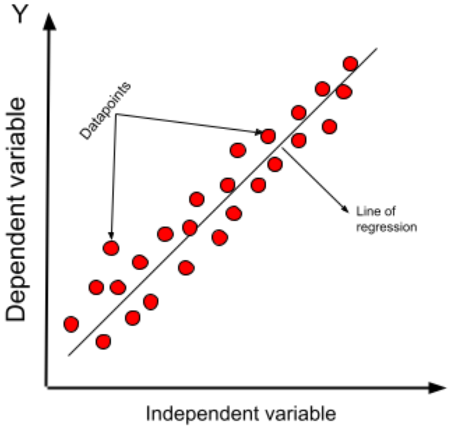


Fig-6: Linear Regression

Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression. In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best fit line for our model.

Hypothesis function for Linear Regression:

$Y1=\theta1 +\theta2.\ x$

While training the model we are given:

x: input training data (univariate – one input variable(parameter))

y: labels to data (supervised learning)

When training the model – it fits the best line to predict the value of y for a given value of x. The model gets the best regression fit line by finding the best θ1 and θ2 values.

θ1: intercept

θ2: coefficient of x

Once we find the best θ1 and θ2 values, we get the best fit line. So, when we are finally using our model for prediction, it will predict the value of y for the input value of x.

Linear regression can be further divided into two types of algorithms. If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression. If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

By achieving the best-fit regression line, the model aims to predict the y value such that the error difference between the predicted value and the true value is minimum. So, it is very important to update the θ1 and θ2 values, to reach the best value that minimizes the error between the predicted y value (pred) and true y value (y).

$$minimize\frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

$$J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

Cost function(J) of Linear Regression is the Root Mean Squared Error (RMSE) between predicted y value (pred) and true y value (y).

3.4.1 Naïve Bayes

The Naïve Bayes algorithm is a supervised learning algorithm, which is based on the Bayes theorem and is used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset.

Naïve Bayes Classifier is one of the simplest and most effective Classification algorithms which helps in building fast machine learning models that can make quick predictions.

It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of the Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, which can be described as:

- **Naïve**: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the basis of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identifying that it is an apple without depending on each other.
- **Bayes**: It is called Bayes because it depends on the principle of Bayes' Theorem

Bayes' Theorem:

- Bayes' theorem is also known as **Bayes' Rule** or **Bayes' law**, which is used to determine the probability of a hypothesis with prior knowledge. It depends on conditional probability.
- The formula for Bayes' theorem is given as:

$$P(A|B)= \frac{P(B|A)P(A)}{P(B)}$$

Where, P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, which can be described as:

- **Naïve**: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the basis of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identifying that it is an apple without depending on each other.

Bayes: It is called Bayes because it depends on the principle of ayes’ Bayes' Theorem

Bayes' Theorem:

- Bayes' theorem is also known as **Bayes' Rule** or **Bayes' law**, which is used to determine the probability of a hypothesis with prior knowledge. It depends on conditional probability.
- The formula for Bayes' theorem is given as:

$$P(A|B)=\frac{P(B|A)P(A)}{P(B)}$$

Were,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

3.4.2 K-nearest neighbor (KNN)

K-nearest neighbor (KNN) algorithm is a type of supervised ML algorithm that can be used for both classifications as well as regression predictive problems. However, it is mainly used for the classification of predictive problems in the industry. The following two properties would define KNN well −

Lazy learning algorithm − KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification.

Non-parametric learning algorithm − KNN is also a non-parametric learning algorithm because it doesn’t assume anything about the underlying data.

Working of KNN Algorithm

K-nearest neighbor (KNN) algorithm uses ‘feature similarity’ to predict the values of new data points which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. We can understand its working with the help of the following steps −

Step 1 − For implementing any algorithm, we need a dataset. So, during the first step of KNN, we must load the training as well as test data.

Step 2 − Next, we need to choose the value of K i.e., the nearest data points. K can be any integer.

Step 3 − For each point in the test data do the following. Calculate the distance between test data and each row of training data with the help of any of the methods namely: Euclidean, Manhattan or Hamming distance. The most commonly used method to calculate distance is Euclidean. Now, based on the distance value, sort them in ascending order. Next, it will choose the top K rows from the sorted array. Now, it will assign a class to the test point based on the most frequent class of these rows. Finally ending it.

3.4.4 Support Vector Machine (SVM)

Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms that are used both for classification and regression. But generally, they are used in classification problems. In the 1960s, SVMs were first introduced but later they got refined in 1990. SVMs have their unique way of implementation as compared to other machine learning algorithms. Lately, they are extremely popular because of their ability to handle multiple continuous and categorical variables.

SVM can be of two types, Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and the classifier used is called a Non-linear SVM classifier.

An SVM model works basically through a representation of different classes in a hyperplane in a multidimensional space. The hyperplane will be generated in an iterative manner by SVM so that the error can be minimized. The goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH).

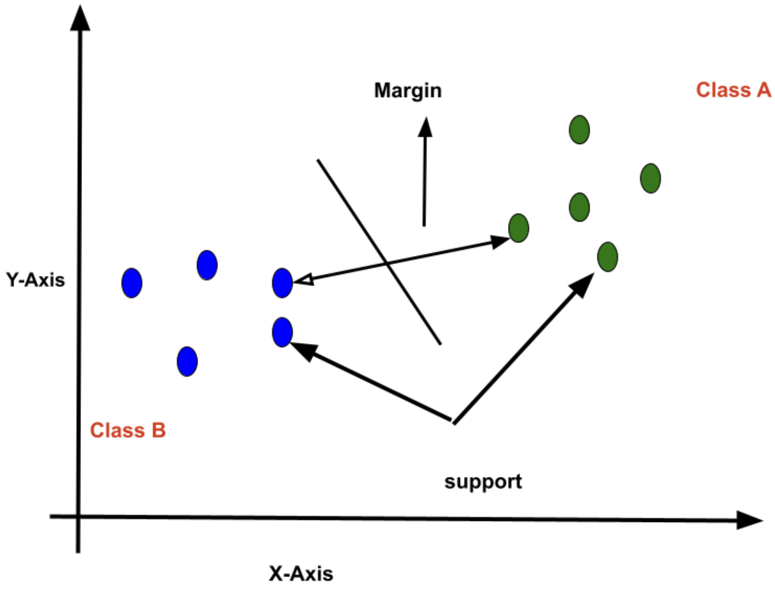


Fig-7: Support Vector Machine

The followings are important concepts in SVM −

- **Support Vectors** − Data Points that are closest to the hyperplane is called support vectors. A separating line will be defined with the help of these data points.
- **Hyperplane** − As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.
- **Margin** − It may be defined as the gap between two lines on the closet data points of different classes. It can be calculated as the perpendicular distance from the line to the support vectors. A large margin is considered as a good margin and a small margin is considered as a bad margin.

The main goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH) and it can be done in the following two steps. First, SVM will generate hyperplanes iteratively that segregates the classes in the best way. Then, it will choose the hyperplane that separates the classes correctly.

3.4.5 Random Forest

Random forest is a supervised learning algorithm that is used for both classifications as well as regression. But it is mainly used for classification problems. As we know a forest is made up of trees and more trees means a more robust forest. Similarly, the Random Forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method that is better than a single decision tree because it reduces over-fitting by averaging the result.

We can understand the working of the Random Forest algorithm with the help of the following steps −

Step 1 − First, start with the selection of random samples from a given dataset.

Step 2 − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.

Step 3 − In this step, voting will be performed for every predicted result.

Step 4 − At last, select the most voted prediction result as the final prediction result.

The following diagram will illustrate how Random Forest works –

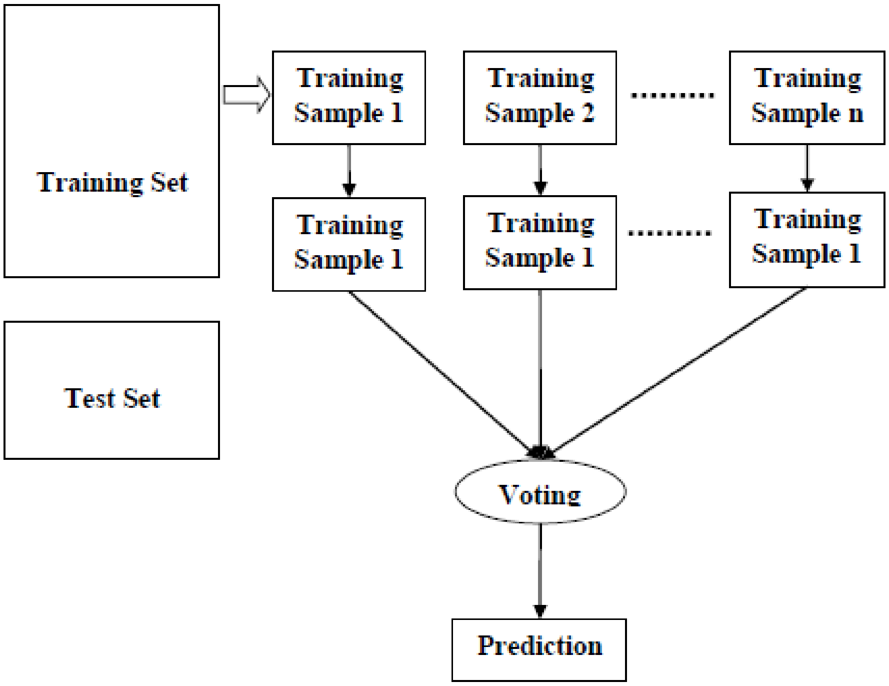


Fig-8: Random Forest Working

For getting high accuracy we used the Random Forest algorithm which gives accuracy which predicates by model and actual outcome of predication in the dataset.

3.5 Testing Method

When reviewing the machine learning models, we usually inspect metrics and plots which summarize model performance over a validation dataset. It is difficult for us to immediately be able to characterize specific model behaviors. It requires additional investigative work. Reporting evaluation metrics are of good practice, but it is not immediately evident how the change occurs. Eventually, it gets difficult to track and prevent the issues often in machine learning models as it tends to fail silently. The real-world dataset is different, so we have to have reported on the behavior of the model to identify where we need the model testing. For Machine learning models there is model evaluation, which summarizes the performance on a validation or test dataset using matrices and plots. Model testing, explicit checks that we expect how our model works.

We would be doing pre-train tests, so we can run them without needing trained parameters, usually, it is difficult to do these tests but doing these helps us better train models without wasting time and prevent errors. Pre-train tests allow us to identify bugs early on.

Here we will split the dataset into 3 sets, Training set, Validation set, Test set.

The training set would be used to train the models over and over again on this same data in our training set as it will learn about the features.

The validation set is a set of data, separate from the training set, that is used to validate our model during training. This validation process helps give information that may assist us with adjusting our hyper parameters. During the training, the model would classify each input from this set as well. The classification will be based only on what is learned about the data it's being trained with. This will not be updated in our actual model result. As the data is separate from the training set the model would get new to validate. This is necessary as a validation set ensures that our model doesn't over fit the data in the training set. During training, if the validation of the model is as good as the results from the training data, we can be more confident that our model wouldn't over fit.

The test set is a set of data that is used to test the model after the model has already been trained. The test set is separate from both the training set and the validation set.

After our model has been trained and validated using our training and validation sets, we will then use our model to predict the output of the unlabeled data in the test set. This set of data should not be labelled so we can see the metrics given during training, for example, the accuracy.

3.5.1 Evaluation metrics

We can evaluate the accuracy of the models using various evaluation metrics such as, Mean Absolute Error, Recall, Precision, F1, Confusion Matrix etc.

Mean Absolute Error is the average of the absolute value of the errors. It gives us the measure of how far the predictions were from the actual output.

Accuracy is a metric for how many of the predictions the model makes are true. The higher the accuracy is, the better. The precision metric marks how often the model is correct when identifying positive results.

Recall, this metric measures the number of correct predictions, divided by the number of results that should have been predicted correctly. It refers to the percentage of total relevant results correctly classified by the algorithms. Recall the ability of a classification model to identify all data points in a relevant class.

A confusion matrix is an $N \times N$ square table, where N is the number of classes that the model needs to classify. Usually, this method is applied to classification where each column represents a label.

Precision is the ability of a classification model to return only the data points in a class. It is the ratio of correctly predict the positive observations to the total predicted positive observations.

F1 score, a single metric that combines recall and precision using the harmonic mean. Evaluating the performance of the models used. The range for F1 Score is [0, 1]. It tells how precise the classifier is.

Chapter 4

Results and Discussion

In this paper we are working with the crop yields, harvested area, temperature, humidity etc. to make the prediction and we have achieved the best score with Random Forest Algorithm which is about 99.97% accurate. Here we will use some evaluation metrics to evaluate the models.

4.1 Analyzation of the model's accuracy

After implementing the algorithm, we have used a few evaluation matrices such as recall, precession, f1 to understand the accuracy of our models.

For Multiple Linear Regression, we evaluated it using Mean Absolute Error (0.0397), R-Squared Score (0.5612), Mean Square Error (0.0128), AUC-ROC Curve **(**0.9603).

For Naïve Bayes algorithm, we evaluated it using Mean Absolute Error (0.0275), Recall Score (0.6923), Precision Score (0.4737), F1 score [0.9858 0.5625], etc. The accuracy for Naïve Bayes came out to be 97.25%.

For Support Vector Machine, we evaluated it Mean Absolute Error (0.0196), Recall Score (0.4118), Precision Score (1.0), F1 score [0.9899 0.5833], etc. The accuracy for Support Vector Machine came out to be 98.04%.

For K-nearest Neighbor, we evaluated it Mean Absolute Error (0. 0078), Recall Score (0.5), Precision Score (0.75), F1 score [0.9960 0.6], etc. The accuracy for K-nearest Neighbor came out to be 99.22%.

For Random Forest, we evaluated it Mean Absolute Error (0. 0031), Recall Score (0.8182), Precision Score (0.75), F1 score [0.9984 0.9], etc. The accuracy for K-nearest Neighbor came out to be 99.69%.

After analyzing all the accuracy scores and the evaluation metrics we can determine that Random Forest Algorithm has the best outcome, which is about 99.69% accurate.

A table consisting of Mean Absolute error, Recall, f1 score, precision etc. from all the algorithms we have applied is given below.

4.2 Comparison of machine learning models

<table>
<tr><th>Algorithm Name</th><th>Mean Absolute Error</th><th>Recall Score</th><th>Precision Score</th><th>F1 Score</th><th>Confusion Matrix</th><th>Accuracy Score</th></tr>
<tr><td><b>Naive Bayes</b></td><td>0.0275</td><td>0.6923</td><td>0.4737</td><td>[.9858 0.5625]</td><td>[487 10]
[4   9]</td><td>0.9725</td></tr>
<tr><td><b>Support Vector Machine</b></td><td>0.0196</td><td>0.4118</td><td>1.0</td><td>[0.9899 0.5833]</td><td>[493 0]
[10 7]</td><td>0.9804</td></tr>
<tr><td><b>Random Forest</b></td><td>0.0031</td><td>0.8182</td><td>1.0</td><td>[0.9984 0.9]</td><td>[626 0]
[2   9]</td><td>0.9969</td></tr>
<tr><td><b>K-nearest Neighbor</b></td><td>0.0078</td><td>0.5</td><td>0.75</td><td>[0.9960 0.6]</td><td>[503 1]
[3   3]</td><td>0.9922</td></tr>
<tr><td rowspan="2"><b>Multiple Linear Regression</b></td><td rowspan="2">0.0397</td><td><b>R-Squared Score</b></td><td><b>Mean Square Error</b></td><td colspan="3" rowspan="2"><b>AUC-ROC Curve</b>
0.9603</td></tr>
<tr><td>0.5612</td><td>0.0128</td></tr>
</table>

SL.	Machine Learning Models	Mean Absolute Error (MAE)
1	Multiple Linear Regression	0.0397
2	Naïve Bayes	0.0275
3	Support Vector Machine	0.0196
4	K-Nearest Neighbour	0.0078
5	Random Forest	0.0031

4.3 Mean Absolute Error Chart

Using the Mean Absolute Error score, we get the following chart.

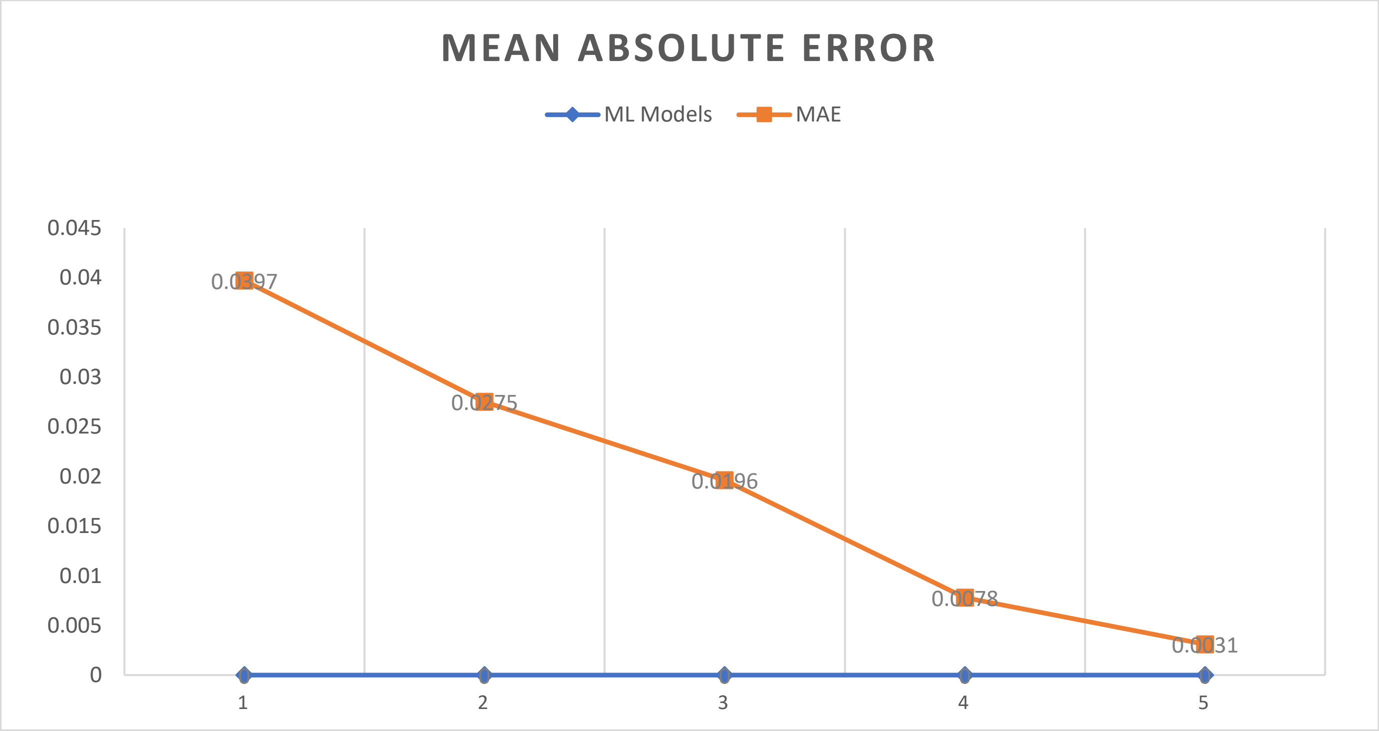


Fig-9: Mean Absolute Error Chart

Chapter 6

Conclusive remarks and limitations

6.1 Conclusion:

In this research we have tried to find out best accuracy of prediction for crop production among different algorithms depending on yields, harvested area and weather from previous hundred-year data. We have found that Random Forest classifier gives the best accuracy. This research will help our farmers to get an idea of production rate from a certain amount of land to minimize the losses and now days it is very much important as the harvesting lands are decreasing day by day. To provide enough food it is important to produce more crops. Then our economy will also be benefited from this research.

6.2 Limitations

In our dataset we can work only in a static way. But it will be really beneficial for users when they can input those data dynamically. We ignore some factors like soil productivity, fertilization and insecticides.

We could not make any surveys in real life because of time limitations. So, we have not been able to collect real feedback of the working experience with the research work. Hopefully we will be able to collect feedback from the farmers in future.

6.3 Future Scope

We wanted to use the Artificial Neural Networks (ANN), Convolution Neural Networks (CNN) algorithm as our future implementation. In the future, we would like to develop apps for the farmers so that they can input data and get results. And also develop an API to private right data to the farmers for their respective data so that they can harvest their crops well.

References

[1] Crop Yield Prediction using Machine Learning Algorithms https://www.ijert.org/crop-yield-prediction-using-machine-learning-algorithms

[2] Crop Yield Prediction Using Random Forest Algorithm for Major Cities in Maharashtra State https://www.researchgate.net/publication/351918010_Crop_Yield_Prediction_Using_Random_Forest_Algorithm_for_Major_Cities_in_Maharashtra_State

[3] An artificial intelligence based approach for increasing agricultural yield https://indjst.org/articles/an-artificial-intelligence-based-approach-for-increasing-agricultural-yield

[4] Crop Yield Prediction Using Linear Support Vector Machine https://ejmcm.com/article_4024_93c3578b3c294e6c760c669ae530f416.pdf

[5] Rice Yield Forecasting Using Support Vector Machine https://www.ijrte.org/wp-content/uploads/papers/v8i4/D7236118419.pdf

[6] Improving Crop Productivity Through a Crop Recommendation System Using Ensembling Technique https://ieeexplore.ieee.org/document/8768790

[7] Agro-Genius: Crop Prediction using Machine Learning https://www.researchgate.net/publication/337783644_Agro-Genius_Crop_Prediction_Using_Machine_Learning

[8] Agricultural Data https://knoema.com/atlas/Bangladesh/Crop-production-index

[9] Weather Data https://power.larc.nasa.gov/data-access-viewer/

[9]https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_support_vector_machine.htm

[10] https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

[11] https://www.javatpoint.com/machine-learning-random-forest-algorithm

[12] https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/

[13] https://www.jeremyjordan.me/testing-ml/