

**MACHINE LEARNING TECHNIQUES FOR CROP YIELD
PREDICTION**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF**

**BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING**

SUBMITTED BY

Name	Univ. Roll No.
Vishal Kumar Pasad	10080120148
Somnath Maji	10080120154
Sourav Singh	10800120155
Abhishek Kumar Chauhan	10800120177

UNDER THE GUIDANCE OF

**Mr. Simanta Hazra
Assistant Professor**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ASANSOL ENGINEERING COLLEGE
AFFILIATED TO
MAULANABULI KALAM AZAD UNIVERSITY OF TECHNOLOGY**

June, 2024

**MACHINE LEARNING TECHNIQUES FOR CROP YIELD
PREDICTION**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF**

**BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING**

SUBMITTED BY

Name	Univ. Roll No.
Vishal Kumar Prasad	10080120148
Somnath Maji	10080120154
Sourav Singh	10800120155
Abhishek Kumar Chauhan	10800120177

UNDER THE GUIDANCE OF

**Mr. Simanta Hazra
Assistant Professor**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ASANSOL ENGINEERING COLLEGE
AFFILIATED TO
MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY**

June, 2024

Contents

Certificate of Recommendation.....	iii
Certificate of Approval.....	iv
Acknowledgement.....	v
Abstract.....	vi
List of Figures.....	vii
List of Tables.....	viii
1. Preface.....	ix
1.1 Introduction.....	x
1.2 Motivation of the project.....	
2. Literature Review.....	xi
2.1 Review of related works	xii
2.2 Crop yield prediction.....	xiii
2.3 ERD.....	xiv
2.4 Overview of datasets.....	xiv
2.5 Methodology.....	
3. Related Theories and Algorithms.....	xv
3.1 Libraries.....	xvi
3.2 Fundamental algorithms.....	
4. Proposed model/algorithm.....	xxii
4.2 Proposed algorithms.....	
5. Simulation Results.....	xxvii
5.1 Experimental setup.....	xxviii
5.2 Source Code.....	xxxv
5.3 Flask framework	xxxvii
5.4 Experimental result.....	xl
5.5 Stimulation result	
6. Discussion and conclusion.....	xl
6.1 Discussion and conclusion.....	
7. References.....	xl
7.1 References	



DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING

ASANSOL ENGINEERING COLLEGE

Vivekananda Sarani, Kanyapur, Asansol, West Bengal – 713305

Certificate of Recommendation

I hereby recommend that the thesis entitled, “**Machine learning techniques for crop yield production**” carried out under my supervision by the group of students listed below may be accepted in partial fulfilment of the requirement for the degree of “Bachelor of Technology in Computer Science and Engineering” of Asansol Engineering College under MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY.

Name	Univ. Roll No.
Vishal Kumar Prasad	10800120148
Somnath Maji	10800120154
Sourav Singh	10800120155
Abhishek Kumar Chauhan	10800120177

Mr. Simanta Hazra

Project Supervisor

Dept. of Computer Science and
Engineering,
Asansol Engineering College,
Asansol-713305

Countersigned:

Dr. Monish Chatterjee

Head of the Department

Dept. of Comp. Sc. & Engg,

Asansol Engineering College,

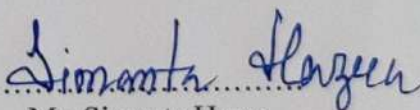
Asansol-713305



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING**
ASANSOL ENGINEERING COLLEGE
Vivekananda Sarani, Kanyapur, Asansol, West Bengal – 713305

Certificate of Approval

The thesis is hereby approved as creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance in the partial fulfilment of the degree for which it has been submitted. It is understood that by this approval the undersigned does not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it is submitted.



Mr. Simanta Hazra
Thesis Supervisor

Dept. of Computer Science and
Engineering,
Asansol Engineering College,
Asansol-713305

Acknowledgement

It is our great privilege to express our profound and sincere gratitude to our Thesis Supervisor, **Mr. Simanta Hazra** for providing us very cooperative and valuable guidance at every stage of the project work being carried out under his supervision. His valuable advice and instructions in carrying out the present study has been a very rewarding and pleasurable experience that has greatly benefited us throughout the period of work.

We would like to convey our sincere gratitude towards Dr. Monish Chatterjee, Head of the Department, Asansol Engineering College for providing us the requisite support for timely completion of our work. We would also like to pay our heartiest thanks and gratitude to all the teachers of the Department of Computer Science and Engineering, Asansol Engineering College for various suggestions being provided in attaining success in our work.

We would like to express our earnest thanks to Mr. Suman Mallick, of CSE Project Lab for his technical assistance provided during our project work.

Finally, I would like to express my deep sense of gratitude to my parents for their constant motivation and support throughout my work.

Vishal Kumar Prasad

Vishal Kumar Prasad

Somnath Maji

Somnath Maji

Sourav Singh

Sourav Singh

Abhishek Kumar Chauhan

Abhishek Kumar Chauhan

Abstract

Fertilizer value updation has a positive practical significance for guiding agricultural production and for notifying the change in market rate of fertilizer to the farmer. The concept of this paper is to implement the crop selection method so that this method helps in solving many agriculture and farmers problems. This improves our Indian economy by maximizing the yield rate of crop production. Different types of land condition. So the quality of the fertilizers are identified using ranking process. By this process the rate of the low quality and high quality fertilizer is also notified. The usage of ensemble of classifiers paves a path way to make a better decision on predictions due to the usage of multiple classifiers. Further, a ranking process is applied for decision making in order to select the classifiers results. This system is used to predict the crop for further.

List of Figures

Fig. 2.2	Flow of proposed crop yield prediction	xii
Fig. 2.3	ERD	viii
Fig. 3.2.1	Independent vs Dependent variables Ex1	xviii
Fig. 3.2.2	Dependent vs Independent variables Ex2	xviii
Fig. 3.2.3	Lasso Regression tutorial	xx
Fig. 3.2.4	Model complexity vs Error	xxii
Fig. 3.2.5	KNN classifier	xxiii
Fig. 3.2.6	Distance function	xxiii
Fig. 3.2.7	Feature vs target	xxiv
Fig. 3.2.8	Decision tree	xxv
Fig. 3.2.9	R-squared Goodness of fit	xxvii
Fig. 5.2	Source code	xxxiv
Fig. 5.3	Output	xxxix
Fig. 5.4.1	Graph Frequency vs Area	xl
Fig. 5.4.2	Yield per Country Graph	xli

List of Tables

Table 2.4 Overview of Dataset

xiv

INTRODUCTION

From ancient period, agriculture is considered as the main and the foremost culture practiced in India. Ancient people cultivate the crops in their own land and so they have been accommodated to their needs. Therefore, the natural crops are cultivated and have been used by many creatures such as human beings, animals and birds. The greenish goods produced in the land which have been taken by the creature leads to a healthy and welfare life. Since the invention of new innovative technologies and techniques the agriculture field is slowly degrading. Due to these, abundant invention people are been concentrated on cultivating artificial products that is hybrid products where there leads to an unhealthy life. Nowadays, modern people don't have awareness about the cultivation of the crops in a right time and at a right place. Because of these cultivating techniques the seasonal climatic conditions are also being changed against the fundamental assets like soil, water and air which lead to insecurity of food. By analyzing all these issues and problems like weather, temperature and several factors, there is no proper solution and technologies to overcome the situation faced by us. In India there are several ways to increase the economical growth in the field of agriculture. There are multiple ways to increase and improve the crop yield and the quality of the crops. Data mining also useful for predicting the crop yield production. Generally, data mining is the process of analyzing data from different perspectives and summarizing it into useful information. Data mining software is an analytical tool that allows users to analyze data from many different dimensions or angles, categorize, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases. The patterns, associations, or relationships among all this data can provide information. Information can be converted into knowledge about historical patterns and future trends. For example, summary information about crop production can help the farmers identify the crop losses and prevent it in future.

Crop yield prediction is an important agricultural problem. Each and Every farmer is always tries to know, how much yield will get from his expectation. In the past, yield prediction was calculated by analyzing farmer's previous experience on a particular crop. The Agricultural yield is primarily depends on weather conditions, pests and planning of harvest operation. Accurate information about history of crop yield is an important thing for making decisions related to agricultural risk management. This research focuses on evolution of a prediction model which may be used to predict crop yield production. The proposed method use data mining technique to predict the crop yield production based on the association rules.

Motivation of the project

There are several applications in the field of agriculture. Some of them are listed below.

- **Crop Selection and Crop Yield Prediction**

To maximize the crop yield, selection of the appropriate crop that will be sown plays a vital role. It depends on various factors like the type of soil and its composition, climate, geography of the region, crop yield, market prices etc. Techniques like Artificial neural networks, K-nearest neighbors and Decision Trees have carved a niche for themselves in the context of crop selection which is based on various factors. Crop selection based on the effect of natural calamities like famines has been done based on machine learning (Washington Okori, 2011). The use of artificial neural networks to choose the crops based on soil and climate has been shown by researchers (Obua, 2011). A plant nutrient management system has been proposed based on machine learning methods to meet the needs of soil, maintain its fertility levels, and hence improve the crop yield (Shivnath Ghosh, 2014). A crop selection method called CSM has been proposed which helps in crop selection based on its yield prediction and other factors (Kumar, 2009).

- **Weather Forecasting**

Indian agriculture mainly relies on seasonal rains for irrigation. Therefore, an accurate forecast of weather can reduce the enormous toil faced by farmers in India including crop selection, watering and harvesting. As the farmers have poor access to the Internet as a result of digital-divide, they have to rely on the little information available regarding weather reports. Up-to-date as well as accurate weather information is still not available as the weather changes dynamically over time. Researchers have been working on improving the accuracy of weather predictions by using a variety of algorithms. Artificial Neural networks have been adopted extensively for this purpose. Likewise, weather prediction based on machine learning technique called Support Vector Machines had been proposed (M.Shashi, 2009). These algorithms have shown better results over the conventional algorithms.

- **Smart Irrigation System**

Farming sector consumes a huge portion of water in India. The levels of ground water are dropping down day-by-day and global warming has resulted in climate changes. The river water for irrigation is a big issue of dispute among many states in India. To combat the scarcity of water, many companies have come up with sensor based technology for smart farming which uses sensors to monitor

the water level, nutrient content, weather forecast reports and soil temperature. EDYN Garden sensor is another example (Gupta, 2016).

RELATED WORK

Agricultural management needs simple and accurate estimation techniques to predict rice yields in the planning process (Ji & Wan, 2007). The necessity of the present study were to: (Washington Okori, 2011) identify whether artificial neural network (ANN) models could effectively predict rice yield for typical climatic conditions of the mountainous region, (Miss.Snehal, 2014) evaluate ANN model performance relative to variations of developmental parameters and (Shivnath Ghosh, 2014) compare the effectiveness of multiple linear regression models with ANN models.

Maize crop forecasting has been done using multilayered feed forward network of ANN (Prajneshu, 2008). They considered maize crop yield data as response variable and total human labour, farm power, fertilizer consumption, and pesticide consumption as predictors.

Generalized Regression Neural Networks (GRNN) method is used for forecasting of agricultural crop production (Chaochong, 2008). They found GRNN to be a good technique for prediction grain production in rural areas. It was reported that GRNN model is suitable for non-linear, multi-objectives and multivariate forecasting.

Evaluation of modified k-Means clustering algorithm in crop prediction is demonstrated by the researcher (Utkarsha P, 2014). Their results and evaluation showed the comparison of modified k-Means over k-Means and-Means++ clustering algorithm and found that the modified k-Means has achieved the maximum number of high quality clusters, correct prediction of crop and maximum accuracy count.

A model was developed for forecasting the yield of the sugarcane in Coimbatore district by using the fortnightly weather variable such as average daily maximum and minimum temperature, relative humidity in the morning and evening and total fortnightly rainfall and the yield data (Priya SRK, 2009).

Time series analysis is a method to analyze time on parametric, series data to extract meaningful statistics and other characteristics of the data. Time series forecasting is a model to predict future values based on previously observed values. New concept of crop yield under average climate conditions was

described and it is used in time series techniques on the past yield data to set up a forecasting model.

CROP YIELD PREDICTION

Data Mining is widely applied to agricultural issues. Data Mining is used to analyze large data sets and establish useful classifications and patterns in the data sets. The overall goal of the Data Mining process is to extract the information from a data set and transform it into understandable structure for further use. This paper analyzes the crop yield production based on available data. The Data mining technique was used to predict the crop yield for maximizing the crop productivity. Figure 1 shows the flow of proposed crop yield prediction.

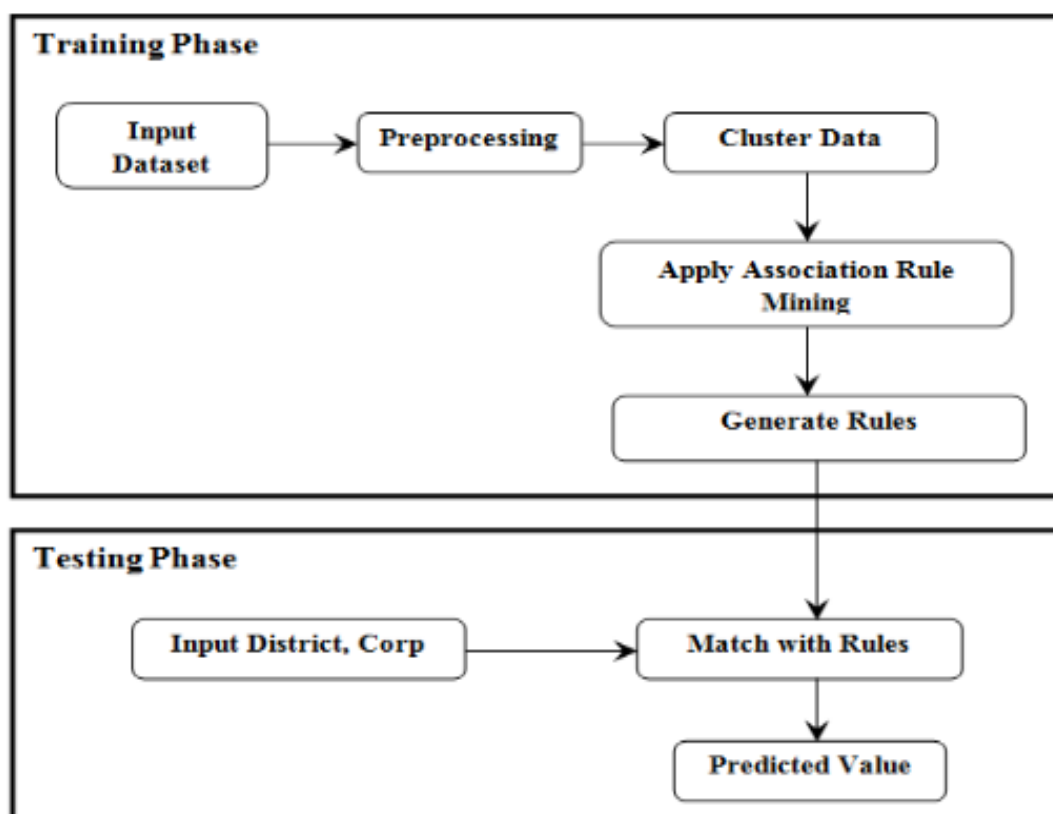


Fig :- 2.2

ERD (Entity Relationship Diagram)

This ERD represents the entities and their relationships involved in predicting crop yield using machine learning.

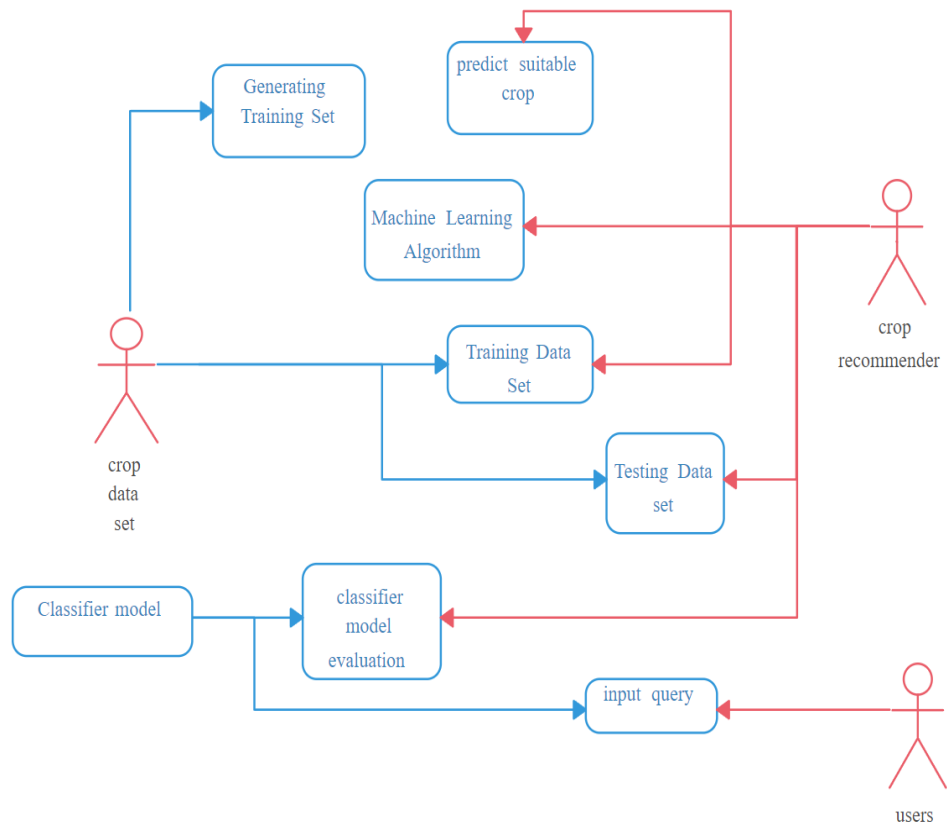


Fig :- 2.3

OVERVIEW OF DATASET

Variable	Description
Area	The total area of agriculture plants region in Hectares and the countries are Albania ,Algeria ,Argentina ,Australia ,Azerbaijan , Bangladesh ,Brazil ,Bulgaria ,Central African Republic,Canada ,Chile , Colombia ,Dominican Republic , France ,Finland ,Germany , India , Turkey etc.
Item	Plants like a Soybeans, potatoes, wheat , maize, Sweet potatoes, Cassava, Rice, paddy Plantains and others Yams Sweet Potatoes etc. from particular area
Year	The data was taken from the year 2000-2012
Hg/ha_yield	Yield of cereals (hectogram per hectare (Hg/Ha))
average_rain_fall_mm_per_year	The average amount of precipitation that falls in a specific location over a year. It's typically measured in millimeters (mm).
pesticides_tonnes	The total amount of pesticides used in a specific area, measured in metric tonnes (tonnes).
avg_temp	The average temperature, likely for a specific location and timeframe.

Table :- 2.4

METHODOLOGY

The proposed methodology contains two phases: Training Phase and Test Phase. In the training phase the data was collected and preprocessed. The pre-processed data was clustered using k-means clustering algorithm. The association rule mining process will apply on clustered data to find the rules. The training phase ends with number of generated rules. In the testing phase, the yield value is predicted based on the generated rules. The work starts with preprocessing step. In this step the collected data was pre-processed. In the preprocessing, some data was removed from the data set. Some of the area was not suitable for crop production. So that data will be removed.

LIBRARIES

1. NumPy (Numerical Python):

What it does: NumPy is a fundamental library for scientific computing in Python. It provides powerful array and matrix manipulation capabilities.

Key features: Efficient multidimensional arrays, linear algebra functions, random number generation, mathematical functions, and tools for integrating with C/C++ code.

2. Pandas:

What it does: Pandas is a high-level library built on top of NumPy, specifically designed for data analysis and manipulation.

Key features: DataFrames (tabular data structures), Series (one-dimensional labeled arrays), data cleaning and transformation tools, time series analysis capabilities, and methods for reading/writing various data formats (CSV, Excel, etc.).

3. Seaborn:

What it does: Seaborn is a visualization library built on top of Matplotlib, offering a high-level interface for creating statistical graphics.

Key features: Easy-to-use functions for creating various plots like scatter plots, heatmaps, violin plots, and distribution plots. It provides a default visual style that improves aesthetics and readability.

4. Matplotlib.pyplot (often shortened to plt):

What it does: Matplotlib is a powerful plotting library for Python. Pyplot is a submodule within Matplotlib that offers a convenient interface for creating various plots.

Key features: Extensive customization options for plots (colors, line styles, labels, legends), ability to create different plot types (line plots, bar graphs, histograms, etc.), and integration with other libraries like NumPy and Pandas.

5. Scikit-learn (often shortened to sklearn):

What it does: Scikit-learn is a comprehensive library for machine learning tasks. It provides a wide range of algorithms for classification, regression, clustering, dimensionality reduction, and model selection.

Key features: Pre-built machine learning models with easy-to-use APIs, data preprocessing tools (scaling, normalization), model evaluation metrics, and tools for hyperparameter tuning.

6. Pickle:

What it does: Pickle is a Python module used for serializing and deserializing objects. It allows you to convert Python objects (like models, data structures) into a byte stream that can be saved to a file or transferred over a network.

Key features: Simplifies saving and loading complex Python objects for later use. This is helpful for saving trained machine learning models or data analysis results for future use.

ALGORITHM

1. Linear Regression :

Linear regression is a type of supervised machine learning algorithm that computes the linear relationship between the dependent variable and one or more independent features by fitting a linear equation to observed data.

When there is only one independent feature, it is known as Simple Linear Regression, and when there are more than one feature, it is known as Multiple Linear Regression.

Similarly, when there is only one dependent variable, it is considered Univariate Linear Regression, while when there are more than one dependent variables, it is known as Multivariate Regression.

Why Linear Regression is Important?

The interpretability of linear regression is a notable strength. The model's equation provides clear coefficients that elucidate the impact of each independent variable on the dependent variable, facilitating a deeper understanding of the underlying dynamics. Its simplicity is a virtue, as linear regression is transparent, easy to implement, and serves as a foundational concept for more complex algorithms.

Linear regression is not merely a predictive tool; it forms the basis for various advanced models. Techniques like regularization and support vector machines draw inspiration from linear regression, expanding its utility. Additionally, linear regression is a cornerstone in assumption testing, enabling researchers to validate key assumptions about the data.

Types of Linear Regression

There are two main types of linear regression:

- **Simple Linear Regression :**

This is the simplest form of linear regression, and it involves only one independent variable and one dependent variable. The equation for simple linear regression is:

$$y = \beta_0 + \beta_1 X$$

where:

Y is the dependent variable

X is the independent variable

β_0 is the intercept

β_1 is the slope

- **Multiple Linear Regression :**

This involves more than one independent variable and one dependent variable. The equation for multiple linear regression is:

$$y = \beta_0 + \beta_1 X + \beta_2 X + \dots \dots \dots \beta_n X$$

where:

Y is the dependent variable

X1, X2, ..., X_p are the independent variables

β_0 is the intercept

$\beta_1, \beta_2, \dots, \beta_n$ are the slopes

The goal of the algorithm is to find the best Fit Line equation that can predict the values based on the independent variables.

In regression set of records are present with X and Y values and these values are used to learn a function so if you want to predict Y from an unknown X this learned function can be used. In regression we have to find the value of Y, So, a function is required that predicts continuous Y in the case of regression given X as independent features.

What is the best Fit Line?

Our primary objective while using linear regression is to locate the best-fit line, which implies that the error between the predicted and actual values

should be kept to a minimum. There will be the least error in the best-fit line.

The best Fit Line equation provides a straight line that represents the relationship between the dependent and independent variables. The slope of the line indicates how much the dependent variable changes for a unit change in the independent variable(s).

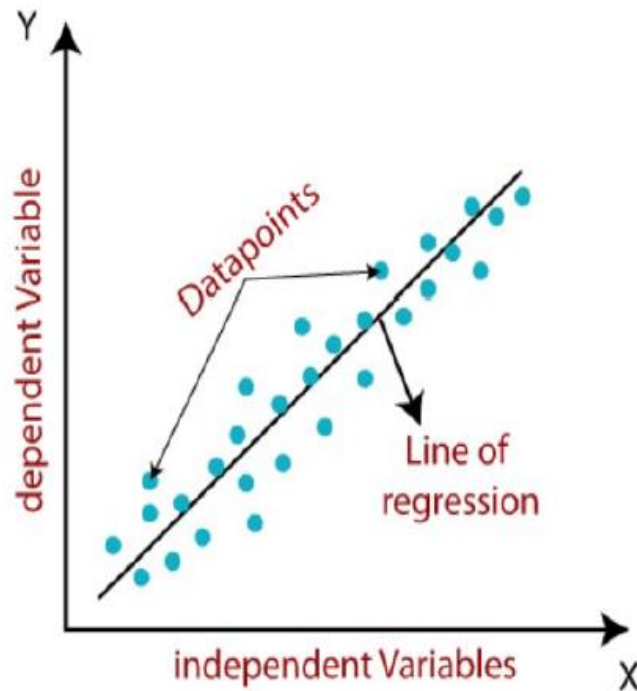


Fig :- 3.2.1

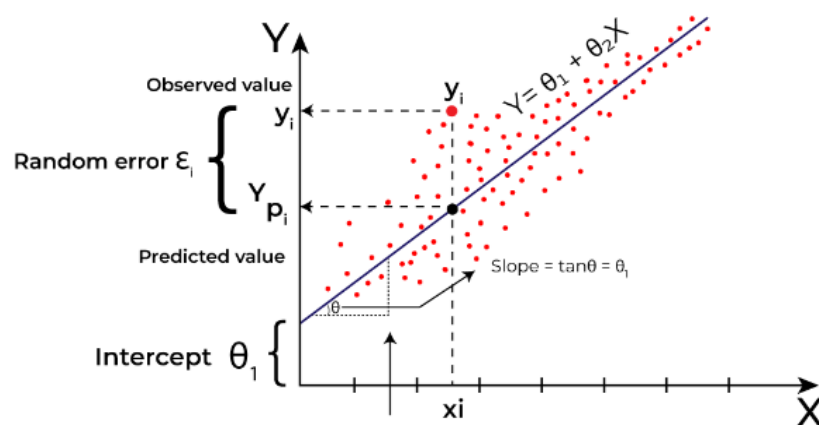


Fig :- 3.2.2

Here Y is called a dependent or target variable and X is called an independent variable also known as the predictor of Y. There are many types of functions or modules that can be used for regression. A linear function is

the simplest type of function. Here, X may be a single feature or multiple features representing the problem.

Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x)). 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.

We utilize the cost function to compute the best values in order to get the best fit line since different values for weights or the coefficient of lines result in different regression lines.

2. Lasso Regression :

Applies L1 regularization, which shrinks some coefficients to zero, potentially leading to feature selection (dropping irrelevant features).

Lasso regression is a regularization technique that applies a penalty to prevent overfitting and enhance the accuracy of statistical models.

These are regularization techniques used with Linear Regression to address overfitting. They penalize the model for having too many large coefficients, leading to a simpler model that generalizes better to unseen data.

Lasso regression—also known as L1 regularization—is a form of regularization for linear regression models. Regularization is a statistical method to reduce errors caused by overfitting on training data. This approach can be reflected with this formula:

$$\hat{w} = \operatorname{argmin}_w \operatorname{MSE}(W) + \|w\|_1$$

Lasso regression is ideal for predictive problems; its ability to perform automatic variable selection can simplify models and enhance prediction accuracy. That said, ridge regression may outperform lasso regression due to the amount of bias that lasso regression introduces by reducing coefficients towards zero. It also has its limitations with correlated features in the data as it arbitrarily chooses a feature to include in the model.

LASSO Regression Tutorial

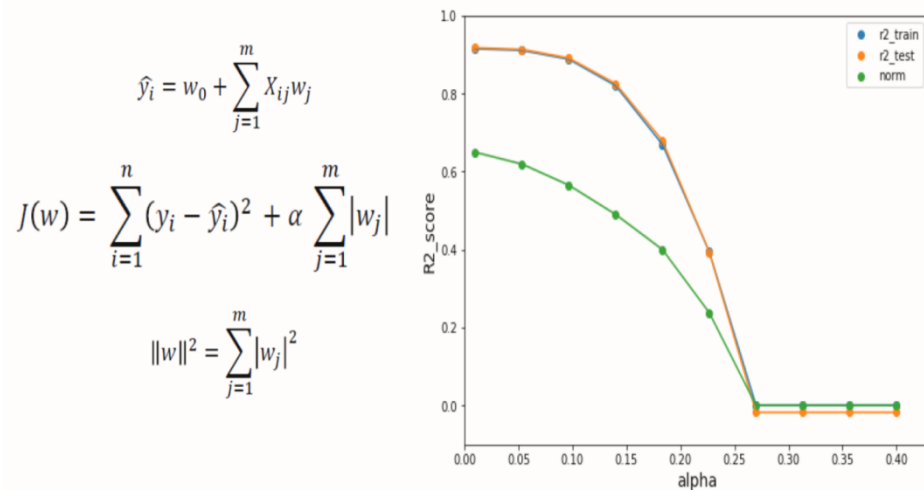


Fig :- 3.2.3

3. Ridge regression :

Applies L2 regularization, which shrinks all coefficients towards zero, but doesn't necessarily drop features.

Ridge regression is a model-tuning method that is used to analyze any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values being far away from the actual values.

The cost function for ridge regression:

$$\text{Min}(\|Y - X(\theta)\|^2 + \lambda \|\theta\|^2)$$

Lambda is the penalty term. λ given here is denoted by an alpha parameter in the ridge function. So, by changing the values of alpha, we are controlling the penalty term. The higher the values of alpha, the bigger is the penalty and therefore the magnitude of coefficients is reduced.

- It shrinks the parameters. Therefore, it is used to prevent multicollinearity.
- It reduces the model complexity by coefficient shrinkage.

- Check out the free course on [regression analysis](#).

Ridge Regression Models :

For any type of regression machine learning model, the usual regression equation forms the base which is written as:

$$Y = XB + e$$

Where Y is the dependent variable, X represents the independent variables, B is the regression coefficients to be estimated, and e represents the errors are residuals.

Once we add the lambda function to this equation, the variance that is not evaluated by the general model is considered. After the data is ready and identified to be part of L2 regularization, there are steps that one can undertake.

Assumptions of Ridge Regressions :

Constant variance, and independence. However, as ridge regression does not provide confidence limits, the distribution of errors to be normal need not be assumed.

Now, let's take an example of a linear regression problem and see how ridge regression if implemented, helps us to reduce the error.

We shall consider a data set on Food restaurants trying to find the best combination of food items to improve their sales in a particular region. The assumptions of ridge regression are the same as those of linear regression: linearity.

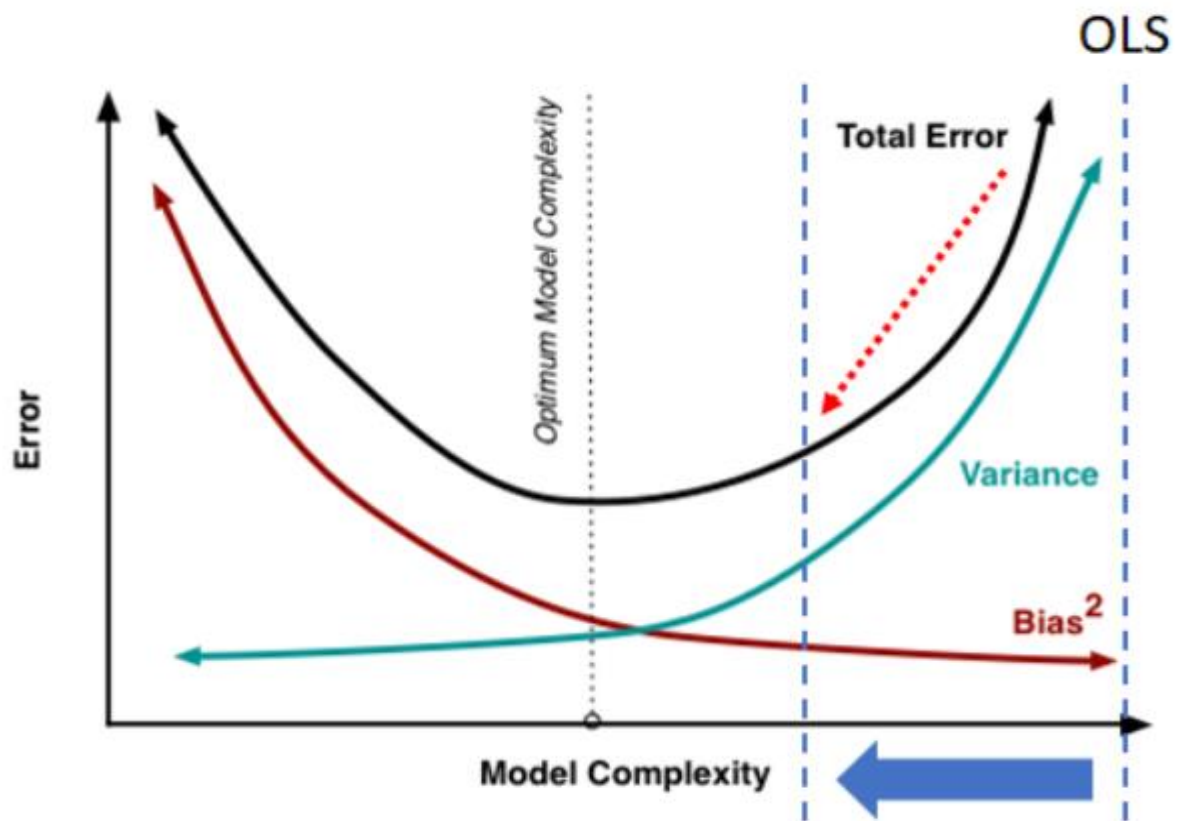


Fig :- 3.2.4

4. KNeighborsRegressor :

K nearest neighbors is a simple algorithm that stores all available cases and predict the numerical target based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique. A simple implementation of KNN regression is to calculate the average of the numerical target of the K nearest neighbors. Another approach uses an inverse distance weighted average of the K nearest neighbors. KNN regression uses the same distance functions as KNN classification.

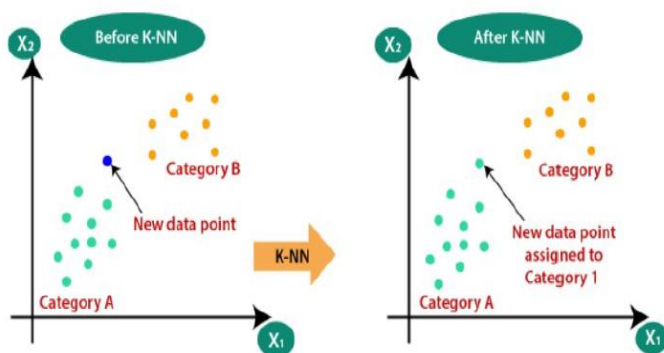
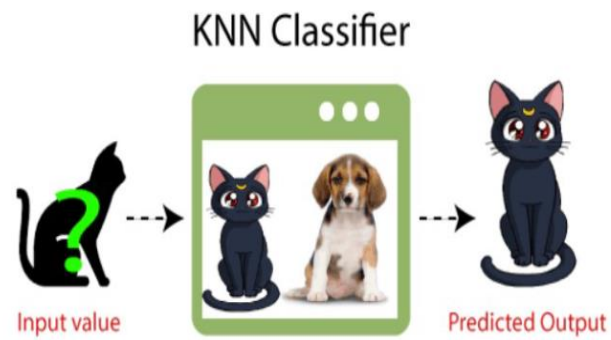


Fig :- 3.1.5

Distance functions

Euclidean	$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$
Manhattan	$\sum_{i=1}^k x_i - y_i $
Minkowski	$\left(\sum_{i=1}^k (x_i - y_i)^q \right)^{1/q}$

Fig :- 3.2.6

The above three distance measures are only valid for continuous variables. In the case of categorical variables you must use the Hamming distance, which is a measure of the number of instances in which corresponding symbols are different in two strings of equal length. The prediction using a single neighbor is just the target value of the nearest neighbor. This non-parametric model predicts the target variable based on the k nearest neighbors (similar data points) in the training data. It's simple to implement but can be computationally expensive for large datasets.

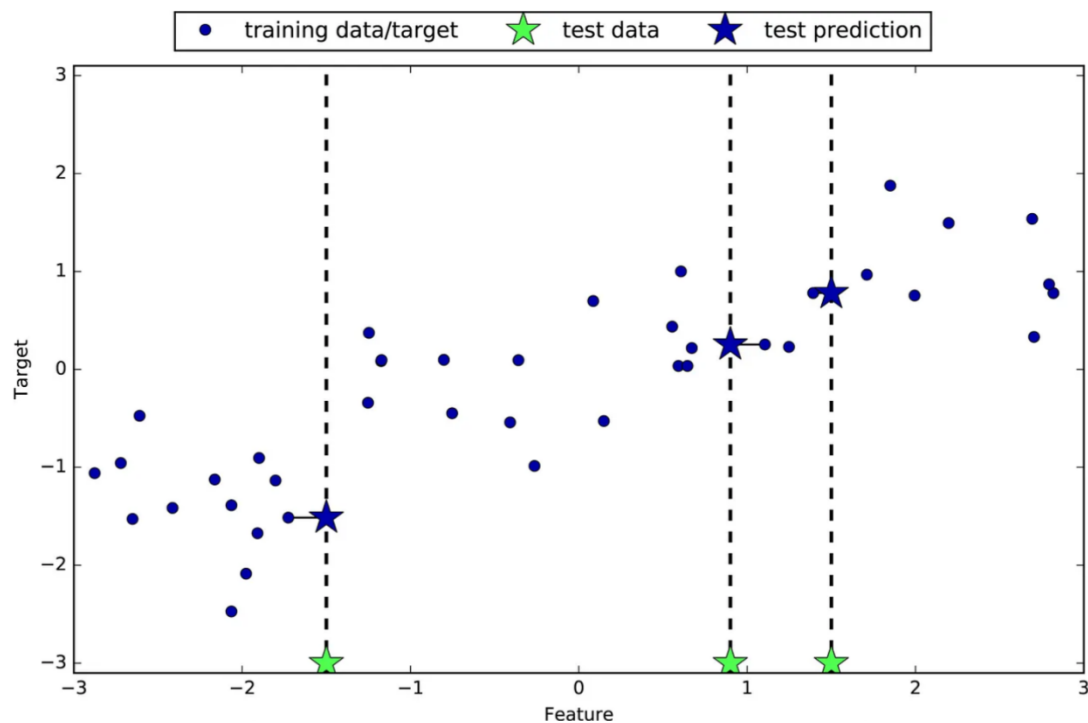


Fig :- 3.2.7

5. DecisionTreeRegressor :

Decision Tree is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs, and utility.

Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables. [1]

The branches/edges represent the result of the node and the nodes have either:

- Conditions [Decision Nodes]
- Result [End Nodes]

The branches/edges represent the truth/falsity of the statement and take makes a decision based on that in the example below which shows a decision tree that evaluates the smallest of three numbers:

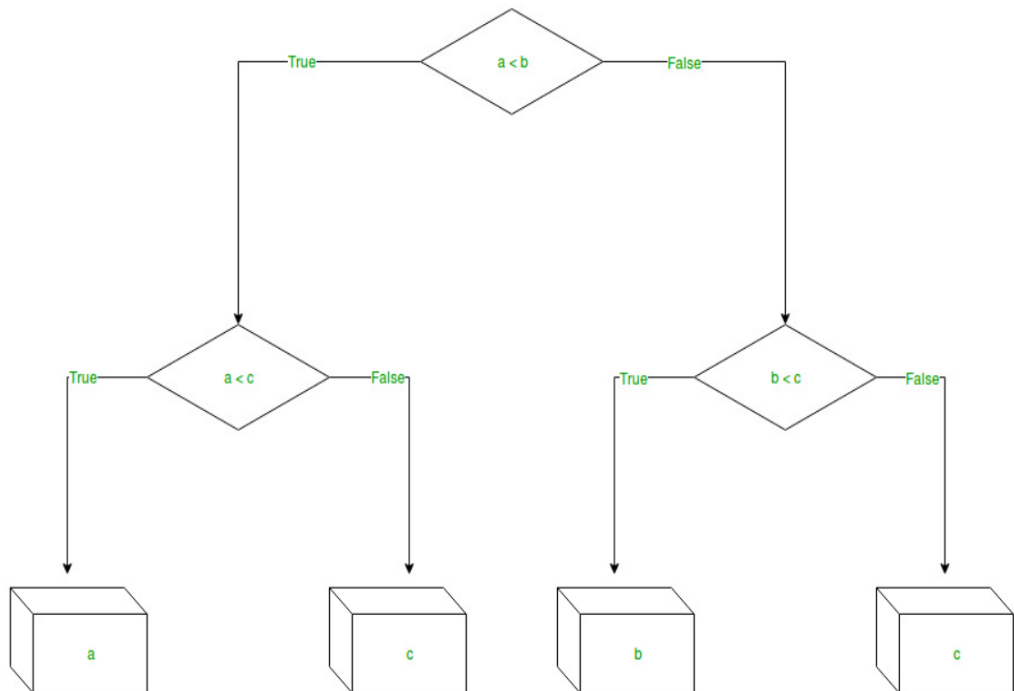


Fig :- 3.2.8

Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values. [3]

Discrete output example: A weather prediction model that predicts whether or not there'll be rain on a particular day.

Continuous output example: A profit prediction model that states the probable profit that can be generated from the sale of a product.

Here, continuous values are predicted with the help of a decision tree regression model.

6. Evaluation Metrics:

- Mean Absolute Error (MAE) :->

This metric measures the average absolute difference between the predicted values and the actual values. It's less sensitive to outliers compared to squared errors.

Mean Absolute Error calculates the average difference between the calculated values and actual values. It is also known as scale-dependent accuracy as it calculates error in observations taken on the same scale. It is used as evaluation metrics for regression models in machine learning. It calculates errors between actual values and values predicted by the model. It is used to predict the accuracy of the machine learning model.[4]

Formula:

$$\text{Mean Absolute Error} = (1/n) * \sum |y_i - x_i|$$

where,

Σ : Greek symbol for summation

y_i : Actual value for the i th observation

x_i : Calculated value for the i th observation

n : Total number of observations

Method 1: Using Actual Formulae :-

Mean Absolute Error (MAE) is calculated by taking the summation of the absolute difference between the actual and calculated values of each observation over the entire array and then dividing the sum obtained by the number of observations in the array.

Method 2: Using sklearn :-

sklearn.metrics module of python contains functions for calculating errors for different purposes. It provides a method named `mean_absolute_error()` to calculate the mean absolute error of the given arrays.

Syntax:

`mean_absolute_error(actual,calculated)`

where

actual- Array of actual values as first argument

calculated – Array of predicted/calculated values as second argument

- R-squared (R^2) :->

R-squared (R^2) is a number that tells you how well the independent variable(s) in a statistical model explain the variation in the dependent variable. It ranges from 0 to 1, where 1 indicates a perfect fit of the model to the data.

R-Squared Goodness Of Fit :

R-square is a comparison of the residual sum of squares (SS_{res}) with the total sum of squares (SS_{tot}). The total sum of squares is calculated by summation of squares of perpendicular distance between data points and the average line. [6]

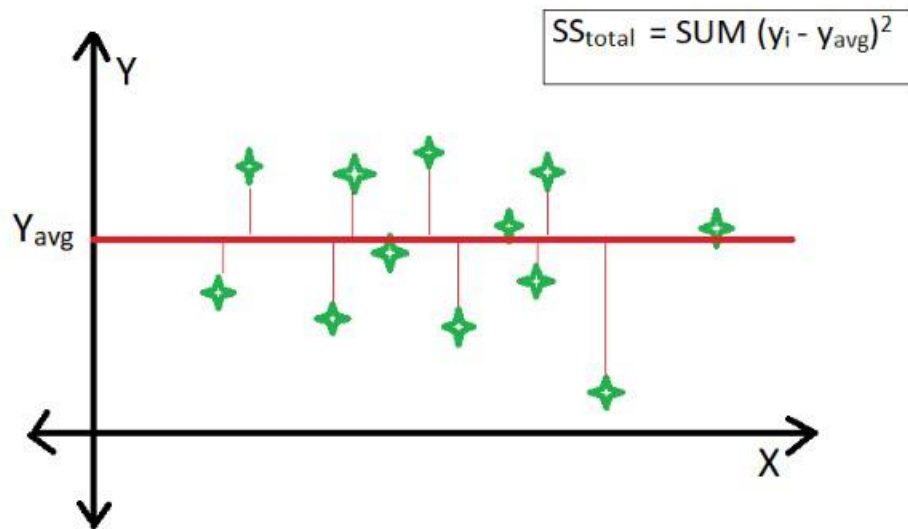


Fig :- 3.2.9

The residual sum of squares is calculated by the summation of squares of perpendicular distance between data points and the best-fitted line.

EXPERIMENT SETUP

R-squared is a statistical measure that indicates how much of the variation of a dependent variable is explained by an independent variable in a regression model. In investing, R-squared is generally interpreted as the percentage of a fund's or security's price movements that can be explained by movements in a benchmark index. An R-squared of 100% means that all movements of a security (or other dependent variable) are completely explained by movements in the index (or whatever independent variable you are interested in).

R-squared represents the proportion of the variance in the dependent variable that is predictable from the independent variables. A value of 1 implies that all the variability in the dependent variable is explained by the independent variables, while a value of 0 suggests that the independent variables do not explain any of the variability. R-squared should be interpreted alongside other statistics and context, as high R-squared values can sometimes be misleading if the model is overfitted.

Whereas correlation explains the strength of the relationship between an independent and a dependent variable, R-squared explains the extent to which the variance of one variable explains the variance of the second variable. So, if the R-squared of a model is 0.50, then approximately half of the observed variation can be explained by the model's inputs.

Formula for R-Squared

$$R^2 = 1 - \left(\frac{\text{Unexplained Variation}}{\text{Total Variation}} \right) / \left(\frac{\text{Unexplained Variation}}{\text{Total Variation}} \right)$$

Source Code

The screenshot shows a Jupyter Notebook interface with the following components:

- Header:** "CropYield-Prediction.ipynb" with a star icon, and a menu bar (File, Edit, View, Insert, Runtime, Tools, Help) with a status "All changes saved".
- Left Panel (Files):** A file explorer showing a directory structure with "sample_data", "dtr.pkl", "preprocessor.pkl", and "yield_df.csv".
- Code Cells:**
 - Cell [1]:** Imports libraries: `import numpy as np # linear algebra`, `import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)`, `import seaborn as sns`, and `import matplotlib.pyplot as plt`.
 - Cell [2]:** `df = pd.read_csv('yield_df.csv')`
 - Cell [3]:** `df.head()`
- Output of Cell [3]:** A preview of the first 5 rows of the dataset:

	Unnamed: 0	Area	Item	Year	kg/ha_yield	average_rain_fall_mm_per_year	pesticides_tonnes	avg_temp
0	0	Albania	Maize	1990	36613	1485.0	121.0	16.37
1	1	Albania	Potatoes	1990	66667	1485.0	121.0	16.37
2	2	Albania	Rice, paddy	1990	23333	1485.0	121.0	16.37
3	3	Albania	Sorghum	1990	12500	1485.0	121.0	16.37
4	4	Albania	Soybeans	1990	7000	1485.0	121.0	16.37
- Next steps:** Two buttons: "Generate code with df" and "View recommended plots".
- Cell [4]:** `df.drop('Unnamed: 0', axis=1, inplace=True)`
- Cell [5]:** `df.shape`
- Output of Cell [5]:** `(28242, 7)`

Files

sample_data

dtr.pkl

preprocessor.pkl

yield_df.csv

+ Code + Text

[6] 1 df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28242 entries, 0 to 28241
Data columns (total 7 columns):
Column Non-Null Count Dtype

0 Area 28242 non-null object
1 Item 28242 non-null object
2 Year 28242 non-null int64
3 hg/ha_yield 28242 non-null int64
4 average_rain_fall_mm_per_year 28242 non-null float64
5 pesticides_tonnes 28242 non-null float64
6 avg_temp 28242 non-null float64
dtypes: float64(3), int64(2), object(2)
memory usage: 1.5+ MB

[7] 1 df.isnull().sum()

Area 0
Item 0
Year 0
hg/ha_yield 0
average_rain_fall_mm_per_year 0
pesticides_tonnes 0
avg_temp 0
dtype: int64

[8] 1 df.duplicated().sum()

2310

[9] 1 df.drop_duplicates(inplace=True)

[10] 1 df.duplicated().sum()

0

Transforming average_rain_fall_mm_per_year

0s completed at 12:49 PM

CropYield-Prediction.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Files

sample_data

dtr.pkl

preprocessor.pkl

yield_df.csv

Transforming average_rain_fall_mm_per_year

In summary, this code identifies the indices of rows in the DataFrame df where the values in the column 'average_rain_fall_mm_per_year' are not numeric strings. These rows can be considered for removal or further processing, depending on the specific use case.

[11] 1 def isStr(obj):
2 try:
3 float(obj)
4 return False
5 except:
6 return True
7 to_drop = df[df['average_rain_fall_mm_per_year'].apply(isStr)].index

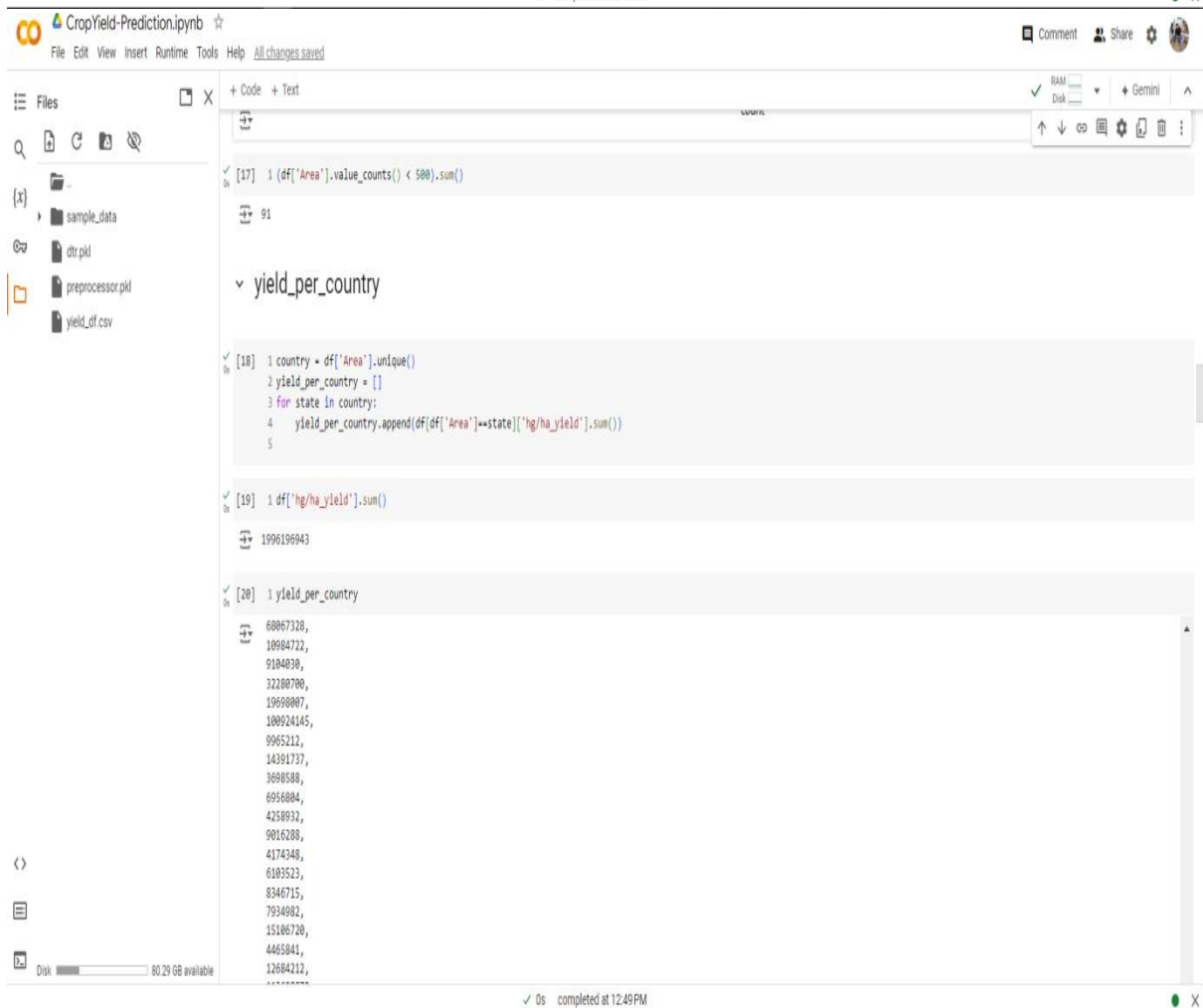
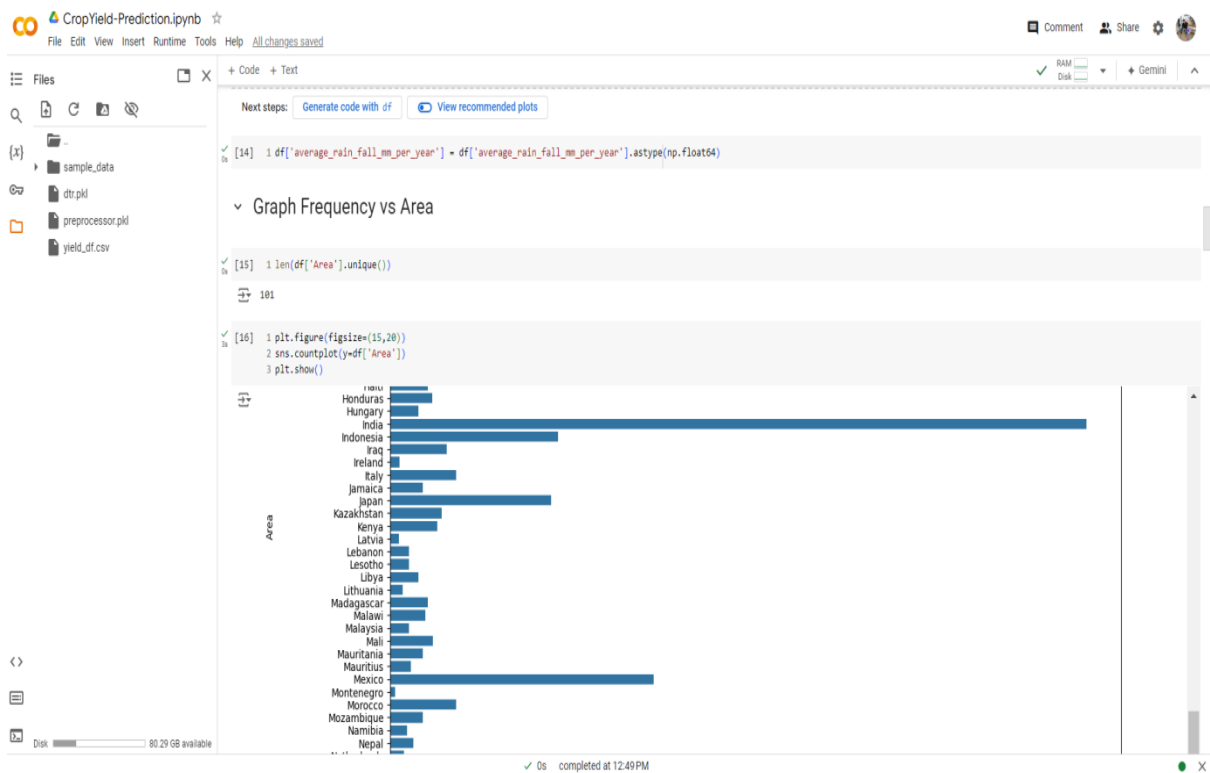
[12] 1 df = df.drop(to_drop)

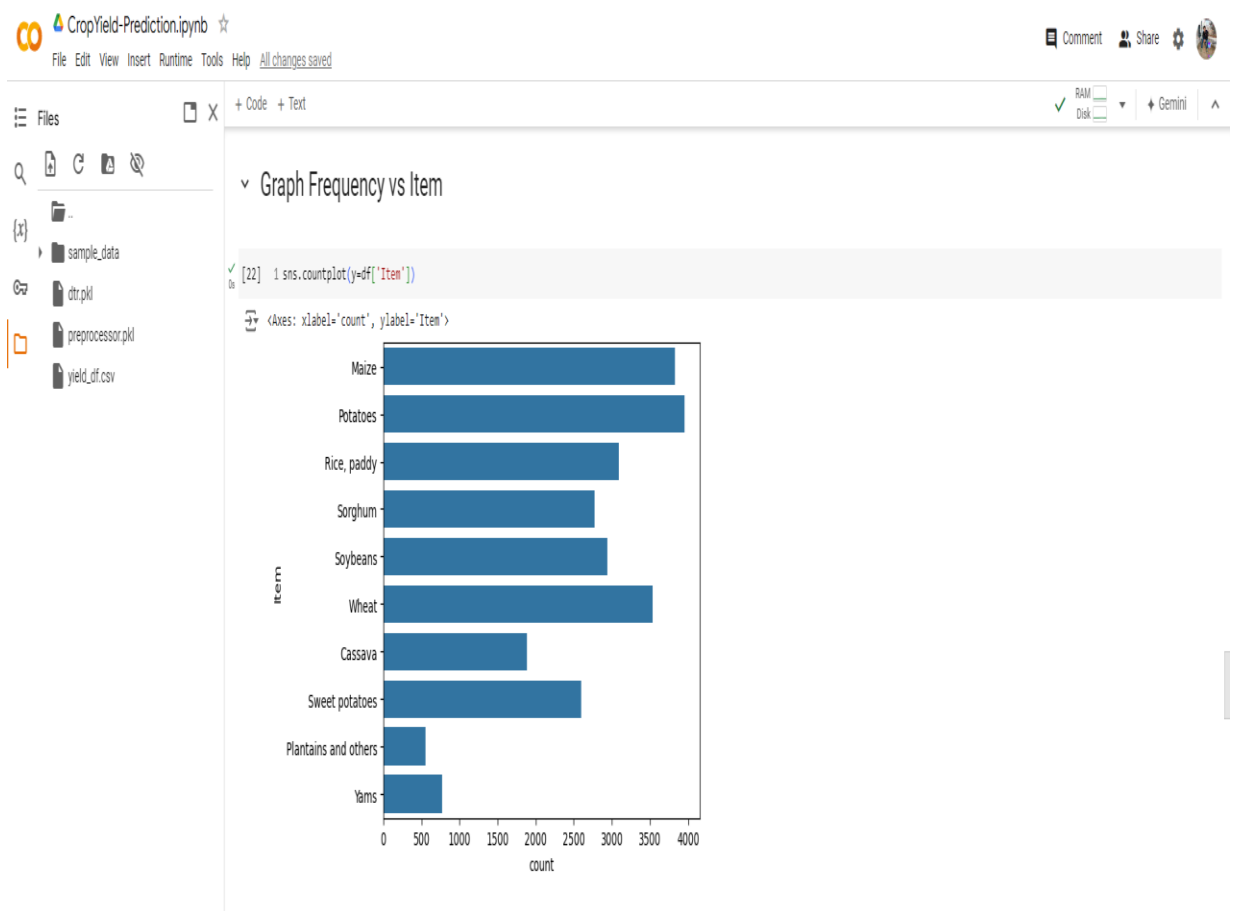
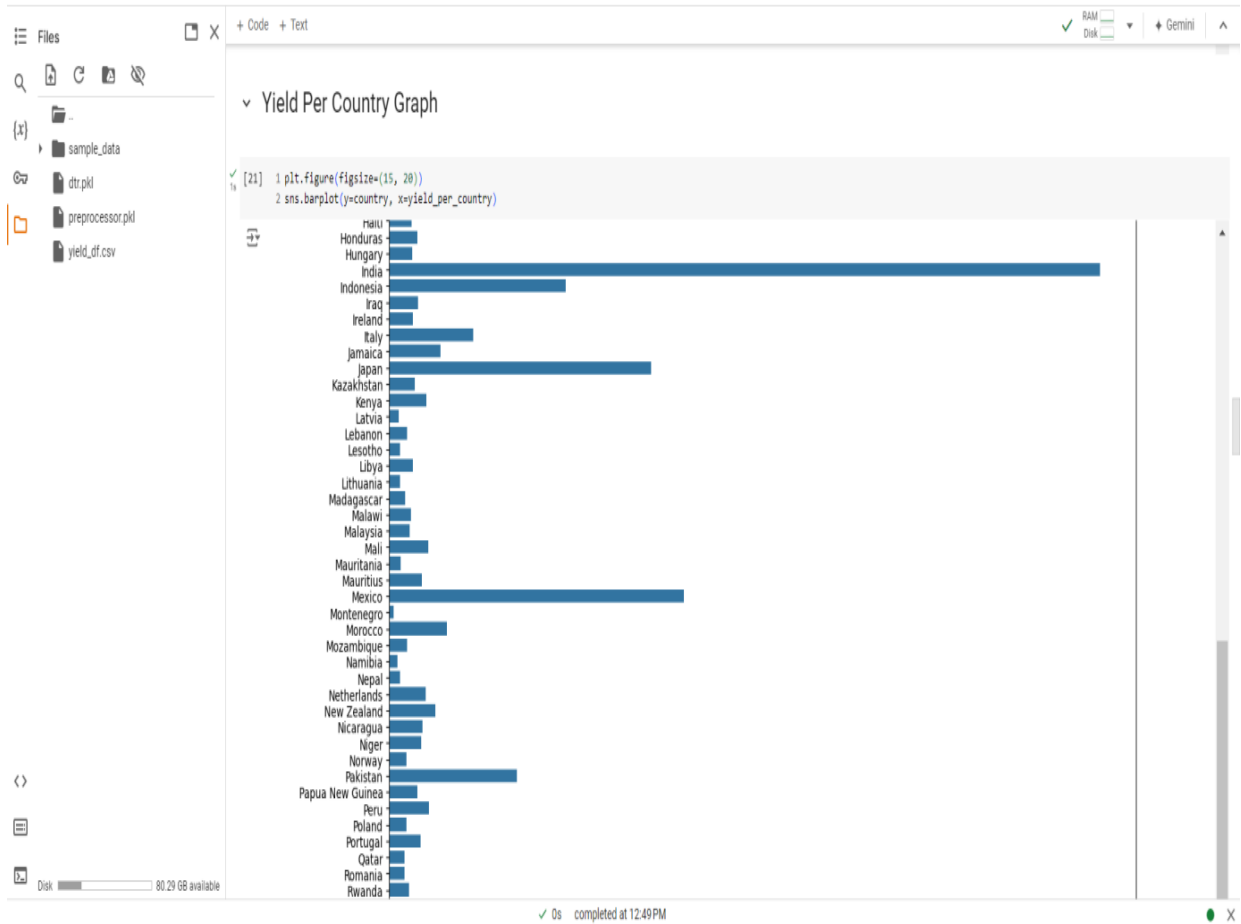
[13] 1 df

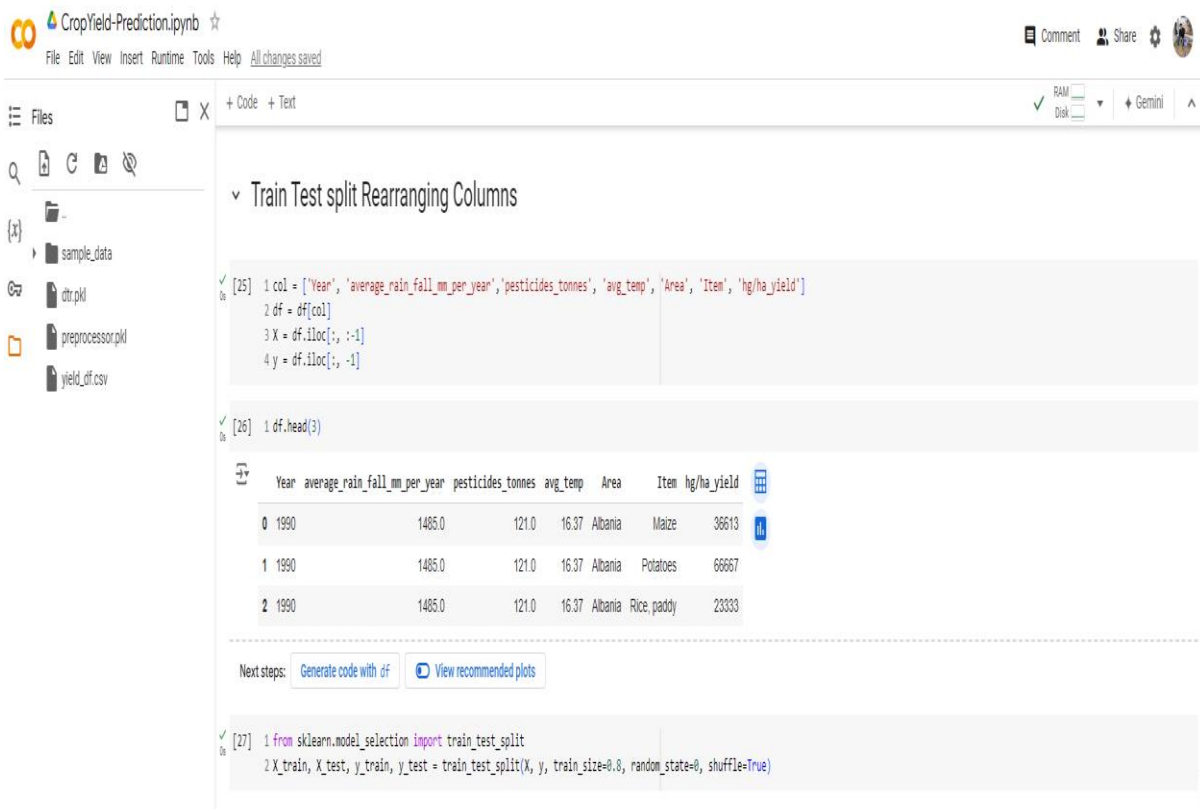
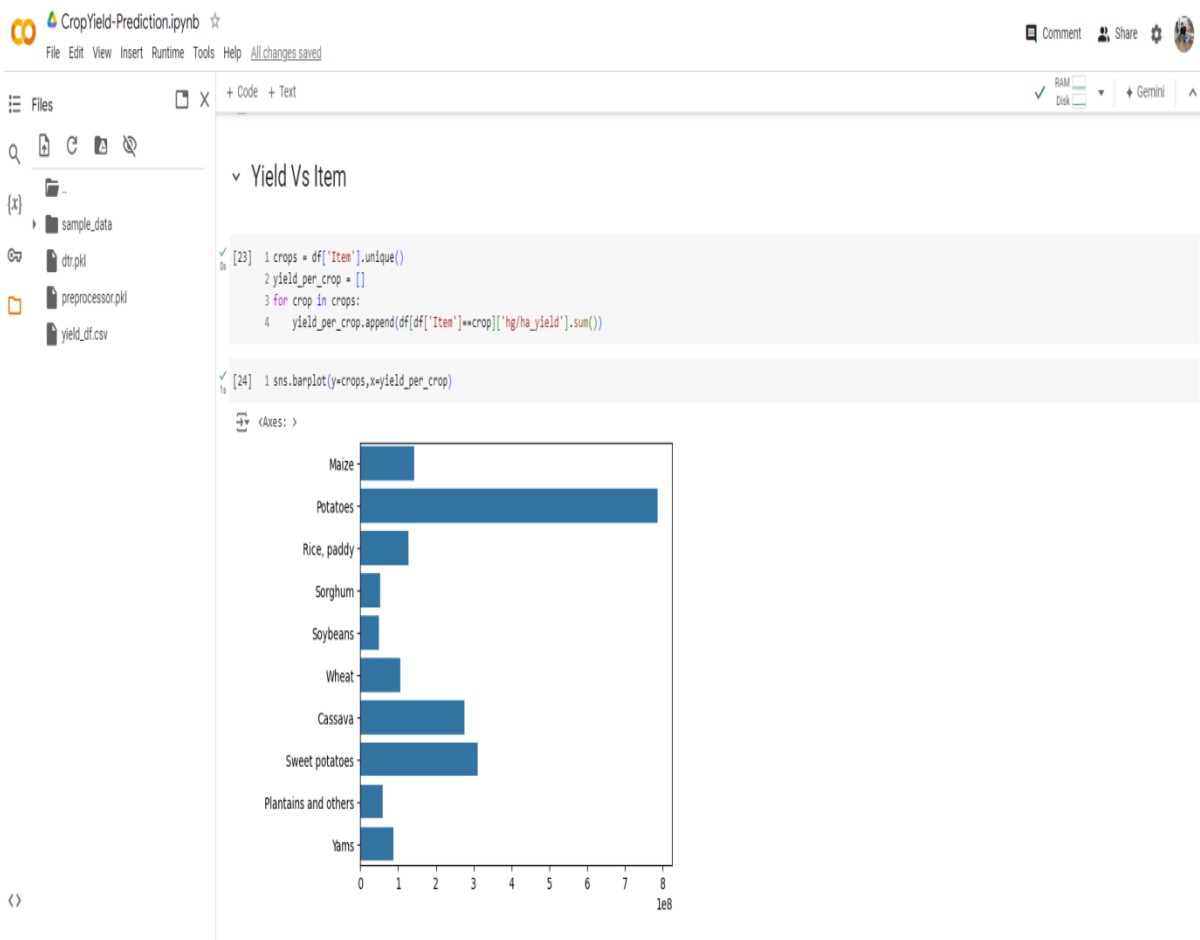
	Area	Item	Year	hg/ha_yield	average_rain_fall_mm_per_year	pesticides_tonnes	avg_temp
0	Albania	Maize	1990	36613	1485.0	121.00	16.37
1	Albania	Potatoes	1990	66667	1485.0	121.00	16.37
2	Albania	Rice, paddy	1990	23333	1485.0	121.00	16.37
3	Albania	Sorghum	1990	12500	1485.0	121.00	16.37
4	Albania	Soybeans	1990	7000	1485.0	121.00	16.37
...
28237	Zimbabwe	Rice, paddy	2013	22581	657.0	2550.07	19.76
28238	Zimbabwe	Sorghum	2013	3066	657.0	2550.07	19.76
28239	Zimbabwe	Soybeans	2013	13142	657.0	2550.07	19.76
28240	Zimbabwe	Sweet potatoes	2013	22222	657.0	2550.07	19.76
28241	Zimbabwe	Wheat	2013	27888	657.0	2550.07	19.76

0s completed at 12:49 PM

xxix







CropYield-Prediction.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Files

sample_data

dtr.pkl

preprocessor.pkl

yield_off.csv

Converting Categorical to Numerical and Scaling the values

```
[28] 1 from sklearn.preprocessing import OneHotEncoder
2 from sklearn.compose import ColumnTransformer
3 from sklearn.preprocessing import StandardScaler
4 ohe = OneHotEncoder(drop='first')
5 scale = StandardScaler()
6
7 preprocessor = ColumnTransformer(
8     transformers = [
9         ('StandardScale', scale, [0, 1, 2, 3]),
10        ('OHE', ohe, [4, 5]),
11    ],
12    remainder='passthrough'
13 )
[29] 1 X_train_dummy = preprocessor.fit_transform(X_train)
2 X_test_dummy = preprocessor.transform(X_test)
[30] 1 preprocessor.get_feature_names_out(col[:-1])
```

array(['StandardScale_Year',
 'StandardScale_average_rain_fall_mm_per_year',
 'StandardScale_pesticides_tonnes', 'StandardScale_avg_temp',
 'OHE_Area_Algeria', 'OHE_Area_Angola', 'OHE_Area_Argentina',
 'OHE_Area_Armenia', 'OHE_Area_Australia', 'OHE_Area_Austria',
 'OHE_Area_Azerbaijan', 'OHE_Area_Bahamas', 'OHE_Area_Bahrain',
 'OHE_Area_Bangladesh', 'OHE_Area_Belarus', 'OHE_Area_Belgium',
 'OHE_Area_Botswana', 'OHE_Area_Brazil', 'OHE_Area_Bulgaria',
 'OHE_Area_Burkina Faso', 'OHE_Area_Burundi',
 'OHE_Area_Cameroon', 'OHE_Area_Canada',
 'OHE_Area_Central African Republic', 'OHE_Area_Chile',
 'OHE_Area_Colombia', 'OHE_Area_Croatia', 'OHE_Area_Denmark',
 'OHE_Area_Dominican Republic', 'OHE_Area_Ecuador',
 'OHE_Area_Egypt', 'OHE_Area_El Salvador', 'OHE_Area_Eritrea',
 'OHE_Area_Estonia', 'OHE_Area_Finland', 'OHE_Area_France',
 'OHE_Area_Germany', 'OHE_Area_Ghana', 'OHE_Area_Greece',
 'OHE_Area_Guatemala', 'OHE_Area_Guinea', 'OHE_Area_Guyana',
 'OHE_Area_Haiti', 'OHE_Area_Honduras', 'OHE_Area_Hungary',
 'OHE_Area_India', 'OHE_Area_Indonesia', 'OHE_Area_Iraq',

RAM

Disk

+ Gemini

0s completed at 12:49 PM

CropYield-Prediction.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Files

sample_data

dtr.pkl

preprocessor.pkl

yield_off.csv

Let's train our model

```
[31] 1 #linear regression
2 from sklearn.linear_model import LinearRegression,Lasso,Ridge
3 from sklearn.neighbors import KNeighborsRegressor
4 from sklearn.tree import DecisionTreeRegressor
5 from sklearn.metrics import mean_absolute_error,r2_score
6
7
8 models = {
9     'lr':LinearRegression(),
10    'lss':Lasso(),
11    'Rid':Ridge(),
12    'Dtr':DecisionTreeRegressor()
13 }
14 for name, md in models.items():
15     md.fit(X_train_dummy,y_train)
16     y_pred = md.predict(X_test_dummy)
17
18     print(f'{name} : mae : {mean_absolute_error(y_test,y_pred)} score : {r2_score(y_test,y_pred)}')
```

lr : mae : 29907.491754632363 score : 0.747312327961836
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_coordinate_descent.py:392: ConvergenceWarning: Objective did not converge. You might want to increase the number of iterations. Dual model = cd.fast.sparse_enet_coordinate_descent(
lss : mae : 29093.99762450549 score : 0.7473261756207235
Rid : mae : 29064.88758387408 score : 0.7473042337440462
Dtr : mae : 3941.058222479275 score : 0.9794424396739345

RAM

Disk

+ Gemini

CropYield-Prediction.ipynb

File Edit View Insert Runtime Tools Help All changes saved

RAM Disk

Files

- sample_data
- dtr.pkl
- preprocessor.pkl
- yield_df.csv

Select model

```
[32] 1 dtr = DecisionTreeRegressor()
      2 dtr.fit(X_train_dummys, y_train)
      3 dtr.predict(X_test_dummys)
```

array([35286., 22814., 19295., ..., 16135., 34879., 79848.])

Predictive System

```
[33] 1 def prediction(Year, average_rain_fall_mm_per_year, pesticides_tonnes, avg_temp, Area, Item):
      2     # Create an array of the input features
      3     features = np.array([[Year, average_rain_fall_mm_per_year, pesticides_tonnes, avg_temp, Area, Item]], dtype=object)
      4
      5     # Transform the features using the preprocessor
      6     transformed_features = preprocessor.transform(features)
      7
      8     # Make the prediction
      9     predicted_yield = dtr.predict(transformed_features).reshape(1, -1)
      10
      11     return predicted_yield[0]
      12
      13 Year = 1990
      14 average_rain_fall_mm_per_year = 1485.0
      15 pesticides_tonnes = 121.00
      16 avg_temp = 16.37
      17 Area = 'Albania'
      18 Item = 'Maize'
      19 result = prediction(Year, average_rain_fall_mm_per_year, pesticides_tonnes, avg_temp, Area, Item)
      20
```

/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but StandardScaler was fitted with feature names warnings.warn(

/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but OneHotEncoder was fitted with feature names warnings.warn(

0s completed at 12:49 PM

CropYield-Prediction.ipynb

File Edit View Insert Runtime Tools Help All changes saved

RAM Disk

Files

- sample_data
- dtr.pkl
- preprocessor.pkl
- yield_df.csv

1 result

array([36613.])

1 Start coding or generate with AI.

Pickle Files

```
[36] 1 import pickle
      2 pickle.dump(dtr, open('dtr.pkl', 'wb'))
      3 pickle.dump(preprocessor, open('preprocessor.pkl', 'wb'))
```

```
[37] 1 import sklearn
      2 print(sklearn.__version__)
```

1.2.2

1 df

	Year	average_rain_fall_mm_per_year	pesticides_tonnes	avg_temp	Area	Item	kg/ha_yield
0	1990	1485.0	121.00	16.37	Albania	Maize	36613
1	1990	1485.0	121.00	16.37	Albania	Potatoes	66667
2	1990	1485.0	121.00	16.37	Albania	Rice, paddy	23333
3	1990	1485.0	121.00	16.37	Albania	Sorghum	12500
4	1990	1485.0	121.00	16.37	Albania	Soybeans	7000
...
28237	2013	657.0	2550.07	19.76	Zimbabwe	Rice, paddy	22581
28238	2013	657.0	2550.07	19.76	Zimbabwe	Sorghum	3066
28239	2013	657.0	2550.07	19.76	Zimbabwe	Soybeans	13142
28240	2013	657.0	2550.07	19.76	Zimbabwe	Sweet potatoes	22222

0s completed at 12:49 PM

Fig :- 5.2

Flask Framework

```
from flask import Flask, request, render_template
import numpy as np
import pickle
import sklearn

print(sklearn.__version__)
# loading models
dtr = pickle.load(open('dtr.pkl', 'rb'))
preprocessor = pickle.load(open('preprocessor.pkl', 'rb'))

# flask app
app = Flask(__name__)

@app.route('/')
def login():
    return render_template('login.html')

@app.route('/Submit')
def index():
    return render_template('index.html')

@app.route("/predict", methods=['POST'])
def predict():
    if request.method == 'POST':
        Year = request.form['Year']
        average_rain_fall_mm_per_year = request.form['average_rain_fall_mm_per_year']
        pesticides_tonnes = request.form['pesticides_tonnes']
        avg_temp = request.form['avg_temp']
        Area = request.form['Area']
        Item = request.form['Item']

        features = np.array([[Year, average_rain_fall_mm_per_year, pesticides_tonnes,
                               avg_temp, Area, Item]],
                               dtype=object)
        transformed_features = preprocessor.transform(features)
        prediction = dtr.predict(transformed_features).reshape(1, -1)

        if(prediction<5000):
            Result = "Yield is 25%"
        elif(prediction>5000 and prediction<10000):
            Result = "Yield is 45%"
        elif(prediction>10000 and prediction<20000):
            Result = "Yield is 65%"
        elif(prediction>20000 and prediction<30000):
            Result = "Yield is 85%"
```

```

elif(prediction>30000):
    Result = "Yield is 100%"
else:
    Result = "Wrong Input"

try:
    return render_template('index.html', prediction=Result)
except ValueError:
    print("Error: cannot add an int and a str")

if __name__ == "__main__":
    app.run(debug=True)

```

Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2. Advantages of using Flask framework are:

- There is a built-in development server and a fast debugger provided.
- Lightweight
- Support Secure cookies
- Templating using Jinja2
- Request dispatching using REST
- Support for unit testing is built-in

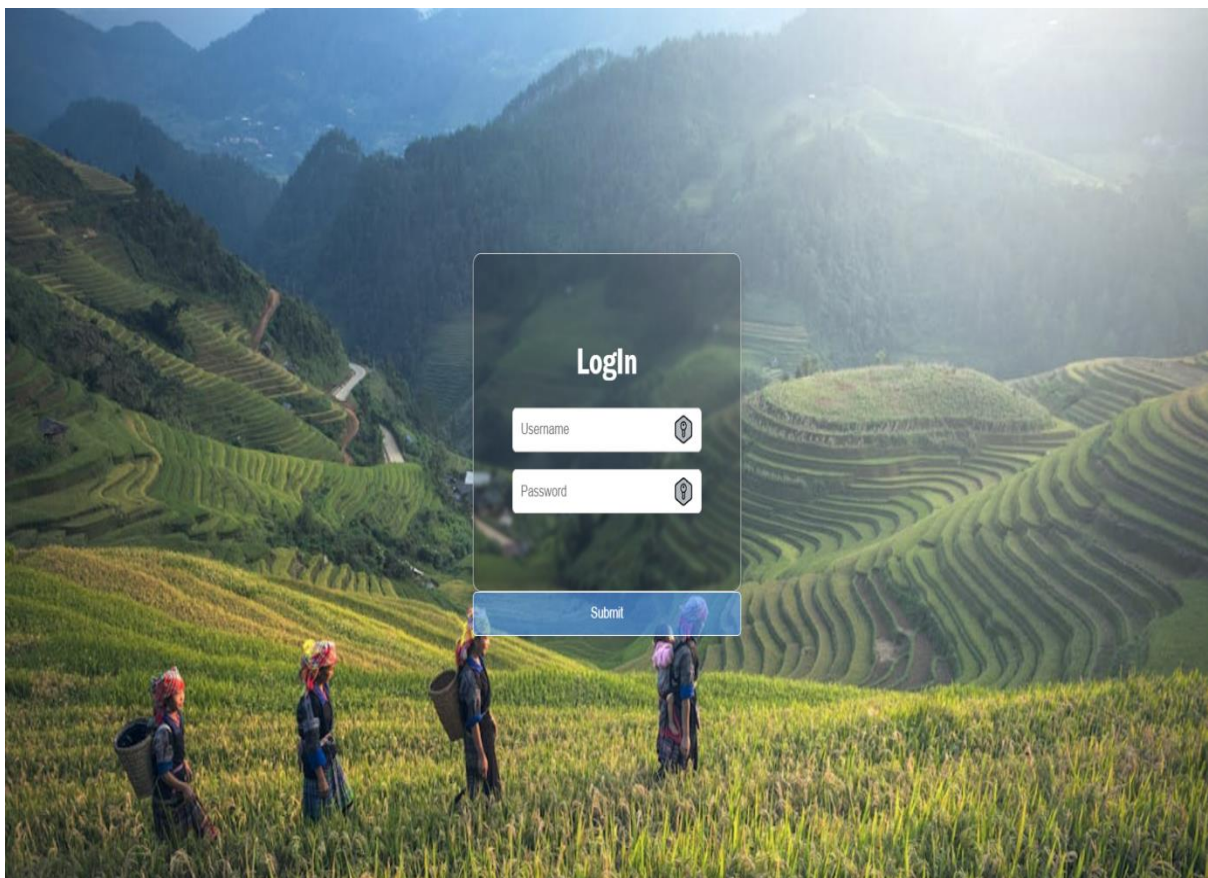
Flask is a backend micro-framework written in Python for the rapid development process. It is famous for its simplicity and independence. It does not need any external library for work, which makes it beginner-friendly, and many people choose this framework. Flask is generally used for building a REST API.

This Flask tutorial is the latest and comprehensive guide designed for beginners and professionals to learn Python Web Framework Flask, which is one of the most popular Python-based web frameworks. Whether you are a beginner or an experienced developer, this tutorial is specially designed to help you learn and master Flask and build your real-world web applications.

This Flask Tutorials covers a wide range of topics from basic concepts such as setup and installation to advanced concepts like user authentication, database integration, and deployment.

EXPERIMENTAL RESULT

```
C:\Windows\py.exe
1.3.2
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
1.3.2
* Debugger is active!
* Debugger PIN: 107-303-159
```



Crop Yield Prediction Per Country

Crop Features Here

Year

Average Rainfall Per Year(in mm)

Pesticides Tonnes

Average Temperature(in Celsius)

Area

Item

Predict

Crop Yield Prediction Per Country

Crop Features Here

Year

2023

Average Rainfall Per Year(in mm)

100

Pesticides Tonnes

28

Average Temperature(in Celsius)

34

Area

India

Item

Maize

Predict

Crop Yield Prediction Per Country

Crop Features Here

Year

Average Rainfall Per Year(in mm)

Pesticides Tonnes

Average Temperature(in Celsius)

Area

Item

Predict

Yield is 65%

Fig :- 5.3

SIMULATION RESULTS

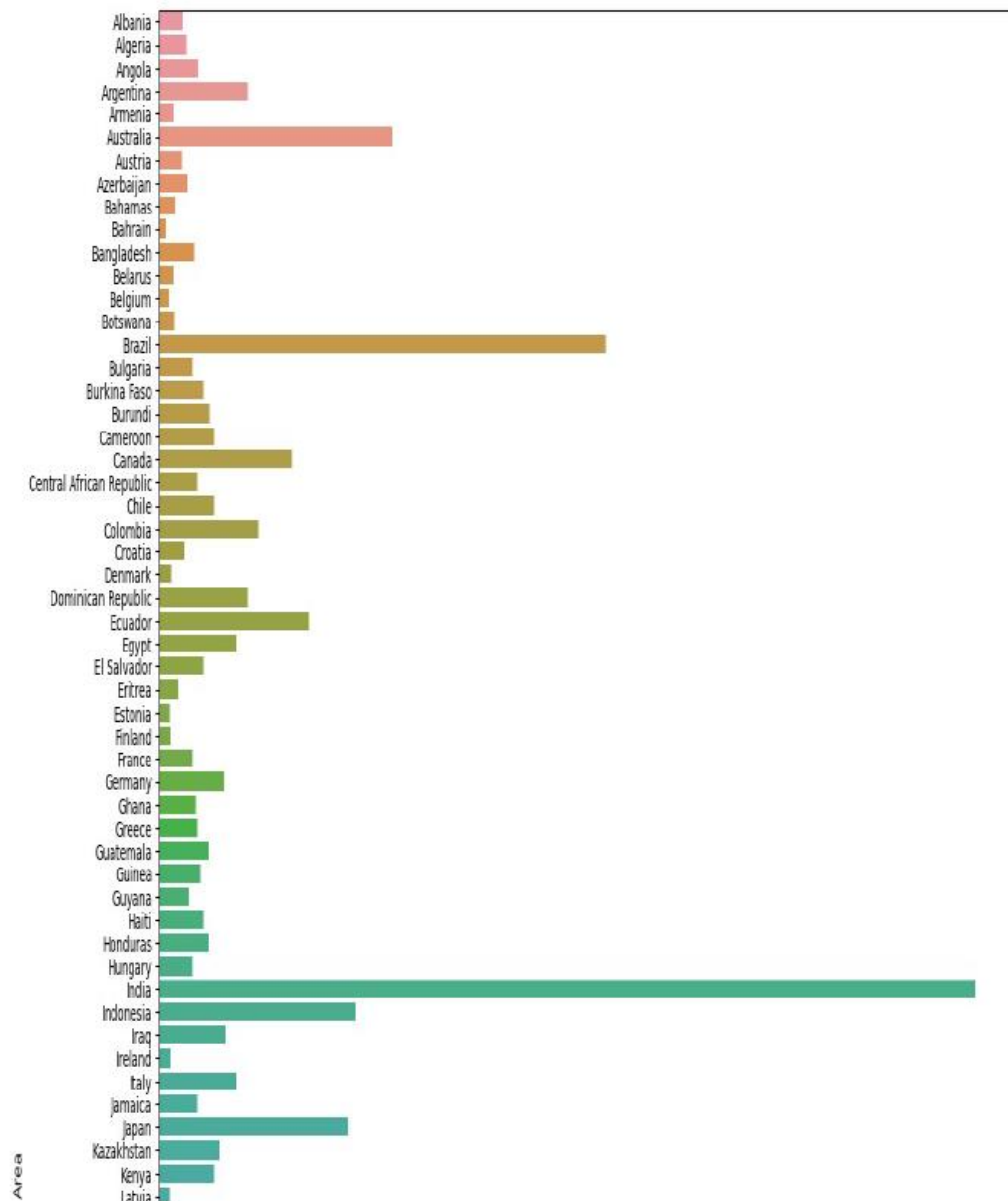


Fig :- 5.4.1

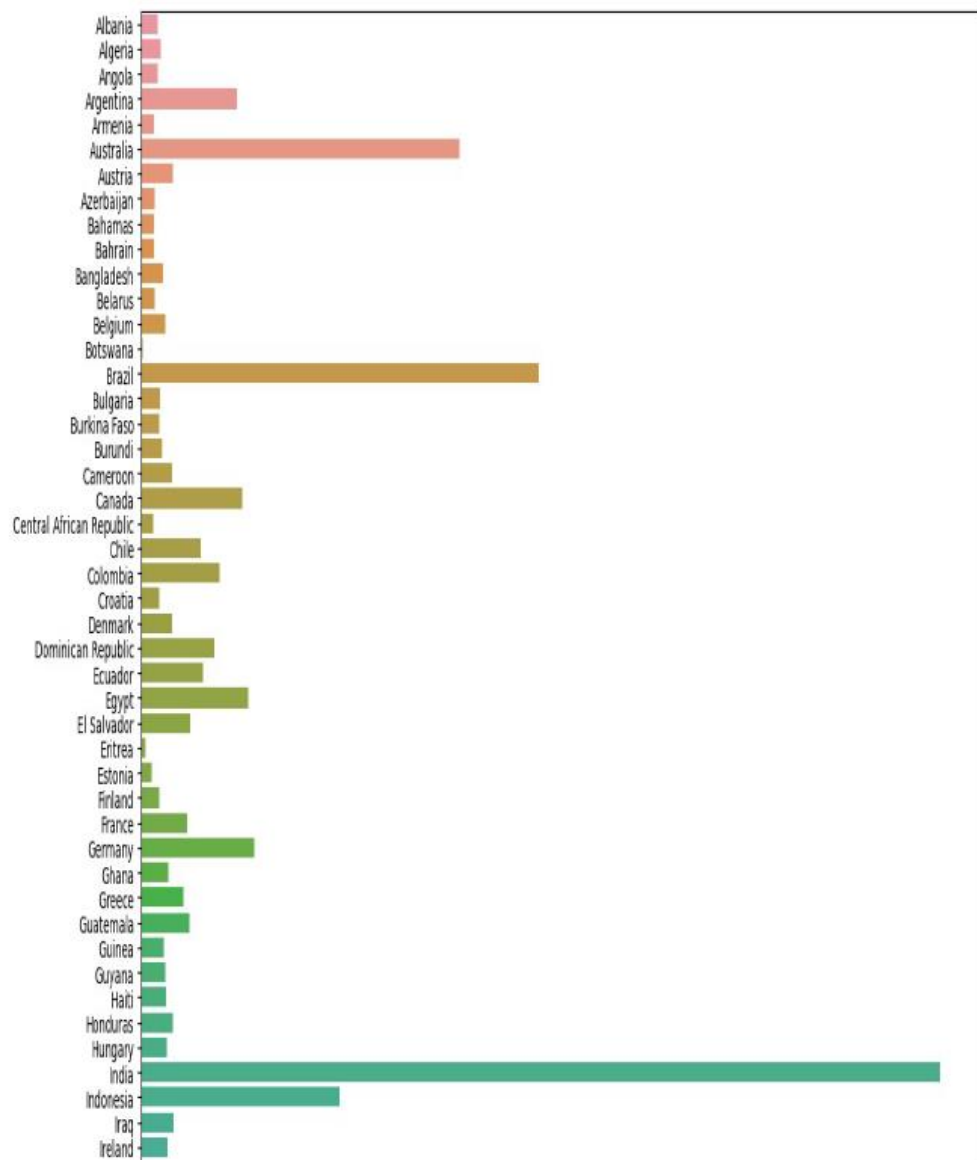


Fig :- 5.4.2

DISCUSSION AND CONCLUSION :

Research highlighted the significance of incorporating machine learning algorithms and IoT sensors in modern agriculture to optimize crop production and reduce waste through informed decision-making. This study identifies the challenges and opportunities associated with integrating these technologies in agriculture. It presents experimental results that demonstrate the impact of changing labels on the accuracy of data analysis algorithms along with accuracy, error values, build, and test time for each classification algorithm. The findings suggest that analyzing wide-ranging data collected from farms, including real-time data from IoT sensors, can enable farmers to make more informed decisions about factors that affect harvest growth. Despite the challenges associated with deploying machine learning in agriculture, our results achieved so far are very promising in that machine learning approaches will become increasingly crucial for production predictions in agriculture in the future. In this experiment, crops were investigated according to general characteristics using different machine learning algorithms, and valuable results were obtained by making predictions in cases where certain crop types are unknown or cannot be easily identified. Our work indicated that appropriate feature selection is critical to achieve better accuracy in machine learning algorithms while analyzing agricultural data. Using the Temperature, Humidity, pH, and Precipitation features in the dataset, it achieved the highest accuracy, reaching 91.05% with Decision Tree Regression and 97.32% with Random Forest. This research provided valuable insights into the potential benefits of these technologies in modern agriculture, and further research and development in this field could help optimize crop production, reduce waste, and improve food security globally.

In future work, more crop data will be evaluated using GPS-based IoT and sensor data from different geographic regions. All these results will be analyzed using a machine learning algorithm. Thus, our data evaluation pool will be established. In addition, different species of the same plant variety will be analyzed separately, and it will be possible to reveal which product is the best type of product among the same species using different machine learning algorithms.

REFERENCE

- [1] Manpreet Kaur, Heena Gulati, Harish Kundra, “Data Mining in Agriculture on Crop Price Prediction: Techniques and Applications”, International Journal of Computer Applications, Volume 99– No.12, August 2014.
- [2] J. Meng, “Research on the cost of agricultural products circulation and its control under the new normal economic development,” Commercial Times, no. 23, pp. 145147, 2016.
- [3] A. Kaloxylou et al., “Farm management systems and the future Internet era,” Comput. Electron. Agricult., vol. 89, pp. 130–144, Nov. 2012.
- [4] N. N. Li, T. S. Li, Z. S. Yu, Y. Rui, Y. Y. Miao, and Y. S. Li, “Factors influencing farmers’ adoption of new technology based on Logistic-ISM model-a case study of potato planting technology in Dingxi City, Gansu Province,” Progress in Geography, vol. 33, no. 4, pp. 542-551, 2014.
- [5] Y. Wang, "A neural network adaptive control based on rapid learning method and its application," Advances In Modeling and Analysis, Vol. 46(3), pp. 27-34,1994.
- [6] Vashisht, S.; Kumar, P.; Trivedi, M.C. Improvised Extreme Learning Machine for Crop Yield Prediction. In Proceedings of the 3rd International Conference on Intelligent Engineering and Management (ICIEM), London, UK, 27–29 April 2022; pp. 754–757.
- [7] OpenAI. New and Improved Content Moderation Tooling. OpenAI. 2022. Available online:<https://openai.com/blog/new-and-improved-content-moderation-tooling/>
- [8] Dean, J. The deep learning revolution and its implications for computer architecture and chip design. In Proceedings of the IEEE International Solid-State Circuits Conference- (ISSCC), San Francisco, CA, USA, 16–20 February 2020.
- [9] Cui, Y.W.; Henrickson, K.; Ke, R.; Pu, Z.; Wang, Y. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. IEEE Trans. Intell. Transp. Syst. 2019, 21, 4883–4894.
- [10] Shahrin, F.; Zahin, L.; Rahman, R.; Hossain, A.J.; Kaf, A.H.; Abdul Malek Azad, A.K.M. Agricultural Analysis and Crop Yield Prediction of Habiganj using Multispectral Bands of Satellite Imagery with Machine Learning. In Proceedings of the 11th International Conference on Electrical and Computer Engineering (ICECE), Dhaka, Bangladesh, 17–19 December 2020; pp. 21–24.

