CAMELLIA INSTITUTE OF TECHNOLOGY

PROJECT-2

CROP YEILD PRODUCTION

By- Sanju Biswas
Sanjana Pashi
Shubhashree Das
Suchitra Sardar

Abstract

India is an agriculture-based nation employing over 50% of the country's workforce. However, despite being called as the backbone of India's economy, the Industry has faced a lot of instability recently. Hence it is only natural that research of various magnitudes is continuously directed towards this field in order to guarantee higher returns in the near future. Keeping in mind the many traditional approaches already prevalent in the agriculture sector which have faced numerous short comings, we would like to propose a competent analysis on the Indian agriculture scenario and how climate change is a affecting it. This is aimed at providing help to the concerned authorities and any future research. Regression algorithms like Ridge Regression, Random Forest etc. have been used to predict the target variable (Production) and have shown great accuracy within the range of 65%-88 % depending on different algorithms. All this data is deployed on an interactive web application using Streamlit. The proposed system will benefit a lot of people as all information related to India Agriculture can be found out at one place and also the prediction of production amount leads to be er planning of resources to be used

Introduction

In the realm of agricultural science and technology, the prediction of crop yields plays a pivotal role in ensuring food security, sustainable farming practices, and informed decision-making for farmers, policymakers, and stakeholders within the agricultural sector. As the global population continues to burgeon, the demand for food production escalates, making it imperative to enhance the efficiency and productivity of crop cultivation. Crop yield prediction, therefore, emerges as a critical component in addressing the challenges posed by unpredictable climatic conditions, resource constraints, and the need for optimized agricultural practices. The advent of advanced technologies, including remote sensing, machine learning, and data analytics, has revolutionized the traditional methods of crop yield estimation. This final year project seeks to delve into the development and implementation of a robust crop yield prediction model that harnesses the power of cutting-edge technologies to provide accurate and timely forecasts. By amalgamating agronomic knowledge with computational techniques, the project aims to contribute to the advancement of precision agriculture, offering farmers actionable insights to optimize resource utilization, mitigate risks, and enhance overall agricultural productivity. Throughout the course of this project, we will explore the intricate interplay between various factors influencing crop yield, such as weather patterns, soil quality, and agronomic practices. Leveraging data-driven approaches, the project endeavors to create a predictive model that goes beyond traditional methodologies, offering a more nuanced and dynamic understanding of the complex dynamics influencing crop growth and yield. The significance of this endeavor extends beyond the confines of academic exploration. The practical implications of an accurate and reliable crop yield prediction model are vast, ranging from aiding farmers in crop planning and resource allocation to informing policymakers in crafting effective agricultural policies. In the face of a changing climate and increasing demands on global food systems, the development of innovative and precise crop yield prediction models becomes indispensable for fostering sustainable agriculture and ensuring food security on a global scale. This project represents a step forward in harnessing technology for the betterment of agriculture, aligning with the broader goal of creating resilient and efficient food production systems to meet the needs of a growing population

Scope of the Project

This projects aims to develop an advanced yield prediction model , integrating data analytics and cutting-edge technologies. It involves collecting diverse data sets on historical crop yields ,weather ,soil quality ,and agronomic practices for a comprehensive understanding of growth factors . The project includes an in-depth Exploratory Data Analysis (EDA) and the development of feature engineering techniques, focusing on incorporating remote sensing data. Machine learning

algorithms will be selected and fine-tuned for optimal crop yield prediction performance. Rigorous validation and testing procedures will assess the model's accuracy and reliability. A user-friendly interface will be designed for practical use by farmers and policymakers, facilitating easy access and interpretation of predictions. Exploration of integration with precision agriculture technologies will enable real-time monitoring and adaptive decision-making in the field. The project emphasizes comprehensive documentation, ethical considerations, and proposes avenues for future research to ensure continuous improvement and adaptation to evolving agricultural needs. In summary, the main purpose of the project is a comprehensive yet generalized analysis which will act as valuable reference material for future and hence would be easy to use and refer to by concerned authorities and users.The project uses various Machine Learning Algorithms and compares them with each other to find the best one suited for the problem statement.This is a low cost and easy implementation technique when compared to other IOT and image processing models coexisting in the market.The project aims to remove the deficiency of good and comprehensive visualization and interpretation of the entire problem statement. Along with that We are proposing the study of a large amount of production data specific to Indian agriculture production scenario.The project is focused mainly on the deployment of the project to an Interactive Data Driven Web Application Using Streamlite Which can then be accessed by anyone.The amount of production levels predicted by the model of the project will help farmers to decide the amount of pesticides and other agriculture tools that they might want to use to meet the predicted production requirement (Rajeshkumar J and Kowsigan, M., 2011),lavanya et al 2020.This will help in major cost cuing and save them from the extra payment losses endured in the process.

Approach , methods and algorithm

Introduction:

In the intricate tapestry of agriculture, the accurate prediction of crop yields stands as a pivotal challenge, with implications extending far beyond the farmstead. Traditional methods, tethered to historical data and manual assessments, often fall short in grasping the dynamic nuances of modern agricultural systems. Recognizing this gap, machine learning algorithms have emerged as beacons of innovation, offering the potential to revolutionize crop yield prediction through a data-driven lens. This study focuses on three key algorithms—Random Forest, Decision Tree, and Support Vector Machines (SVM)—as integral components of precision agriculture, aiming to enhance the accuracy and efficiency of forecasting methods. The Random Forest algorithm, renowned for its ensemble learning approach, amalgamates the strengths of multiple decision trees to create a robust and adaptable predictive model. Decision Trees, with their intuitive and interpretable structure, offer insights into the hierarchical factors influencing crop yields. Meanwhile, Support Vector Machines, as powerful classifiers, bring forth their ability to discern both linear and non-linear relationships within complex agricultural datasets. To fortify the reliability of these predictive models, the study incorporates cross-validation techniques. This systematic validation methodology involves the iterative division of datasets into subsets for training and testing, ensuring the models' capacity to generalize to unseen data and guarding against overfitting. The amalgamation of advanced algorithms and rigorous validation

methods holds the promise of not only enhancing crop yield predictions but also steering agricultural decision-making toward a more sustainable, resilient, and food-secure future.

METHODOLOGY

Overview

We are focused on identifying climate change patterns and its effects in the overall agriculture productivity of India.We are proposing the study of a large amount of production data specific to Indian agriculture production and have tried to predict the future possible effects on agriculture productivity as well.

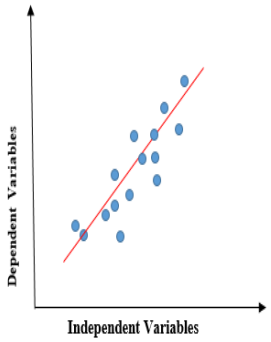
Algorithms

We have tried to predict the production of crops specific to the Indian subcontinent based on features like average Temperature, average Rain, State Name, Crop Type and Area to be cultivated.These features have been selected as these are easy to understand and find. These values can be inserted by any user with no technical knowledge and get the desired output easily. We have tried to cover all of the crops that are grown in India. This does add a lot of disappearances and outliner which leads to a lower accuracy. This trade o comes with its own benefits and downfalls. So, they have been taken care of in this project and every aspect of it has been analyzed. Since the target or dependent variable is the Production column, which has continuous values this means we need to go for algorithms that can predict continuous values.Thus, we have gone with regression algorithms for predictions. Regression is a popular technique used for many Machine Learning Projects.

ALGORITHMS USED

Linear Regression Algorithm

Linear regression is a quiet and simple statistical regression method used for predictive analysis and shows the relationship between the continuous variables. Linear regression shows the linear relationship between the independent variable (X-axis) and the dependent variable (Y-axis), consequently called linear regression. If there is a single input variable (x), we call such linear regression simple linear regression. If there are more than one input variable, we call it multiple linear regression. The linear regression model gives a sloped straight line describing the relationship within the variables.



The above graph presents the linear relationship between the dependent variable and independent variables. When the value of x (independent variable) increases, the value of y (dependent variable) is likewise increasing. The red line is referred to as the best fit straight line. Based on the given data points, we try to plot a line that models the points the best.

To calculate best-fit line linear regression uses a traditional slope-intercept form.

$$y = mx+b \quad \Longrightarrow \quad y = a_0+a_1x$$

- y= Dependent Variable.
- x= Independent Variable.
- a0= intercept of the line.
- a1 = Linear regression coefficient.

Need of a Linear regression

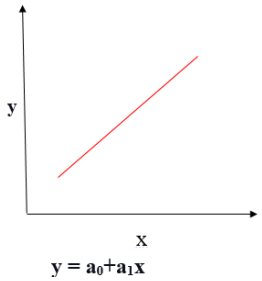
As mentioned above, Linear regression estimates the relationship between a dependent variable and an independent variable. Let's understand this with an easy example:

Let's say we want to estimate the salary of an employee based on year of experience. You have the recent company data, which indicates that the relationship between experience and salary. Here year of experience is an independent variable, and the salary of an employee is a dependent variable, as the salary of an employee is dependent on the experience of an employee. Using this insight, we can predict the future salary of the employee based on current & past information.

A regression line can be a Positive Linear Relationship or a Negative Linear Relationship.

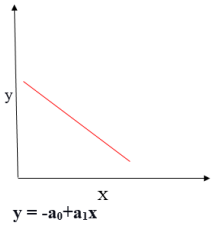
Positive Linear Relationship

If the dependent variable expands on the Y-axis and the independent variable progress on X-axis, then such a relationship is termed a Positive linear relationship.



Negative Linear Relationship

If the dependent variable decreases on the Y-axis while the independent variable increases on the X-axis, we refer to this relationship as a negative linear relationship.



The goal of the linear regression algorithm is to get the best values for a0 and a1 to find the best fit line. The best fit line should minimize the error between predicted values and actual values, ensuring that the error is minimized.

Random Forest Algorithm

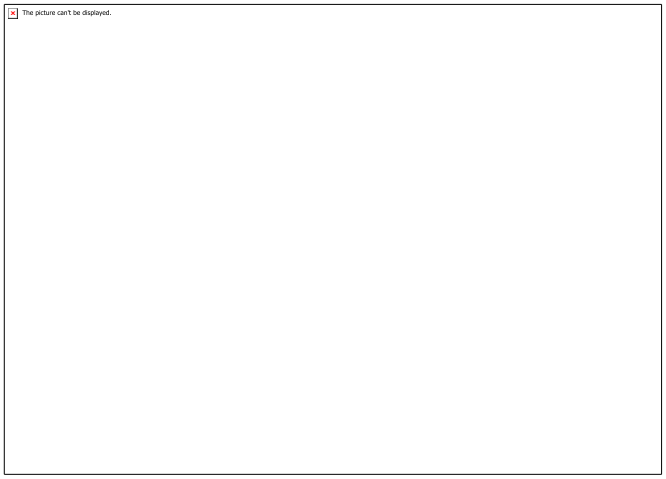
A Random Forest is a collection of decision trees that work together to make predictions. In this article, we'll explain how the Random Forest algorithm works and how to use it.

Understanding Intuition for Random Forest Algorithm

Random Forest algorithm is a powerful tree learning technique in Machine Learning to make predictions and **then we do voting of all the tress to make prediction**. They are widely used for classification and regression task.

- It is a type of classifier that uses many decision trees to make predictions.
- It takes different random parts of the data-set to train each tree and then it combines the results by averaging them. This approach helps improve the accuracy of predictions. Random Forest is based on ensemble learning.

Imagine asking a group of friends for advice on where to go for vacation. Each friend gives their recommendation based on their unique perspective and preferences (decision trees trained on different subsets of data). You then make your final decision by considering the majority opinion or averaging their suggestions (ensemble prediction).



As explained in image: Process starts with a dataset with rows and their corresponding class labels (columns).

- Then - Multiple Decision Trees are created from the training data. Each tree is trained on a random subset of the data (with replacement) and a random subset of features. This process is known as **bagging** or **bootstrap aggregating**.
- Each Decision Tree in the ensemble learns to make predictions independently.
- When presented with a new, unseen instance, each Decision Tree in the ensemble makes a prediction.

The final prediction is made by combining the predictions of all the Decision Trees. This is typically done through a majority vote (for classification) or averaging (for regression).

Key Features of Random Forest

- **Handles Missing Data**: Automatically handles missing values during training, eliminating the need for manual imputation.
- Algorithm ranks **features based on their importance in making predictions** offering valuable insights for feature selection and interpretability.
- **Scales Well with Large and Complex Data** without significant performance degradation.
- Algorithm is versatile and can be applied to both classification tasks (e.g., predicting categories) and regression tasks (e.g., predicting continuous values).

How Random Forest Algorithm Works?

The random Forest algorithm works in several steps:

- Random Forest builds **multiple decision trees using random samples of the data**. **Each tree is trained on a different subset of the data which makes each tree unique**.
- When creating each tree the **algorithm randomly selects a subset of features or variables to split the data rather than using all available features at a time. This adds diversity to the trees.**
- Each decision tree in the forest **makes a prediction based on the data it was trained on. When making final prediction random forest combines the results from all the trees.**
    - For classification tasks the final prediction is decided by a majority vote. This means that the category predicted by most trees is the final prediction.

- For regression tasks the final prediction is the average of the predictions from all the trees.

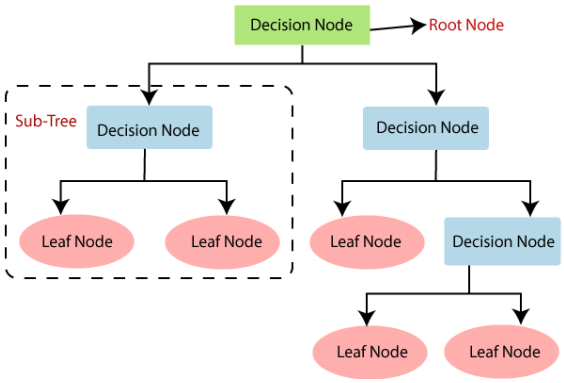
- The **randomness in data samples and feature selection helps to prevent the model from overfitting making the predictions more accurate and reliable.**

Assumptions of Random Forest

- **Each tree makes its own decisions**: Every tree in the forest makes its own predictions without relying on others.
- **Random parts of the data are used**: Each tree is built using random samples and features to reduce mistakes.
- **Enough data is needed**: Sufficient data ensures the trees are different and learn unique patterns and variety.
- **Different predictions improve accuracy**: Combining the predictions from different trees leads to a more accurate final results.

Decision Tree Algorithm

- Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where **internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome.**
- In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- **It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.**
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the **CART algorithm,** which stands for **Classification and Regression Tree algorithm.**
- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.
- Below diagram explains the general structure of a decision tree:



Why use Decision Trees?

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:

- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree-like structure.

Decision Tree Terminologies

· **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
· **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
· **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
· **Branch/Sub Tree:** A tree formed by splitting the tree.
· **Pruning:** Pruning is the process of removing the unwanted branches from the tree.
· **Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes.

How does the Decision Tree algorithm Work?

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

- **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
- **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
- **Step-3:** Divide the S into subsets that contains possible values for the best attributes.
- **Step-4:** Generate the decision tree node, which contains the best attribute.
- **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

Support Vector Regression

Support vector regression (SVR) is a type of support vector machine (SVM) that is used for regression tasks. It tries to find a function that best predicts the continuous output value for a given input value.

SVR can use both linear and non-linear kernels. A linear kernel is a simple dot product between two input vectors, while a non-linear kernel is a more complex function that can capture more intricate patterns in the data. The choice of kernel depends on the data's characteristics and the task's complexity.

In scikit-learn package for Python, you can use the ***'SVR'*** class to perform SVR with a linear or non-linear ***'kernel'***. To specify the kernel, you can set the kernel parameter to ***'linear'*** or ***'RBF'*** (radial basis function).

Concepts related to the Support vector regression (SVR):

There are several concepts related to support vector regression (SVR) that you may want to understand in order to use it effectively. Here are a few of the most important ones:

- **Support vector machines (SVMs):** SVR is a type of support vector machine (SVM), a supervised learning algorithm that can be used for classification or regression tasks. SVMs try to find the hyperplane in a high-dimensional space that maximally separates different classes or output values.
- **Kernels:** SVR can use different types of kernels, which are functions that determine the similarity between input vectors. A linear kernel is a simple dot product between two input vectors, while a non-linear kernel is a more complex function that can capture more intricate patterns in the data. The choice of kernel depends on the data's characteristics and the task's complexity.
- **Hyperparameters:** SVR has several hyperparameters that you can adjust to control the behavior of the model. For example, the **'C'** parameter controls the trade-off between the insensitive loss and the sensitive loss. A larger value of **'C'** means that the model will try to minimize the insensitive loss more, while a smaller value of C means that the model will be more lenient in allowing larger errors.
- **Model evaluation:** Like any machine learning model, it's important to evaluate the performance of an SVR model. One common way to do this is to split the data into a training set and a test set, and use the training set to fit the model and the test set to evaluate it. You can then use metrics like mean squared error (MSE) or mean absolute error (MAE) to measure the error between the predicted and true output values.

Gradient Boost Algorithm

Prediction models are one of the most commonly used machine learning models. Gradient boosting Algorithm in machine learning is a method standing out for its prediction speed and accuracy, particularly with large and complex datasets. This

algorithm has produced the best results from Kaggle competitions to machine learning solutions for business. It is a boosting method, and I have talked more about it in this article. It is refer it as Stochastic Gradient Boosting Machine or GBM Algorithm. In this article, I will discuss the math intuition behind the Gradient boosting algorithm.

Gradient boosting is a machine learning ensemble technique that sequentially combines the predictions of multiple weak learners, typically decision trees. It aims to improve overall predictive performance by optimizing the model's weights based on the errors of previous iterations, gradually reducing prediction errors and enhancing the model's accuracy. This technique is most commonly used for **linear regression**. Errors play a major role in any machine learning algorithm. There are two main types of errors: bias error and variance error. The gradient boost algorithm helps us minimize the model's bias error. The main idea behind this algorithm is to build models sequentially, and these subsequent models try to reduce the errors of the previous model. But how do we do that? How do we reduce the error? Build a new model on the errors or residuals of the previous model.

When the target column is continuous, we use a Gradient Boosting Regression; when it is a classification problem, we use a Gradient Boosting Classifier. The only difference between the two is the "Loss function". The objective is to minimize this loss function by adding weak learners using gradient descent. Since it is based on the++ loss function, for regression problems, we'll have different loss functions like Mean squared error (MSE), and for classification, we will have different functions, like log-likelihood.

Dataset

Data for the project has been collected from data.gov.in , a site by the government of India consisting of many other social datasets.The main dataset consists of Crop Production from 1997 till 2015 (table 1.) state wise with every crop included in the dataset. The dataset has around 2-3 Lakh rows.

Sl.No	Column	Non-Null	Count	Dtype
0	State_Name	240691	Non-null	object
1	District_Name	240691	Non-null	object
2	Crop_year	240691	Non-null	int64
3	Season	240691	Non-null	object
4	Crop	240691	Non-null	object
5	Area	240691	Non-null	float64
6	Production	240691	Non-null	float64

The environmental factors considered for the project are rainfall and temperature, as through the literature analysis it was evident that these two factors are the most important since they affect the Indian crops most.The data for average rainfall and temperature was taken from the same website. Both the datasets are from 1901 to 2015. All three datasets were merged with respect to the Year column .The final dataset after merging has values till 2015.

Cleaning and Removing Outliers

Dataset aer merging has to be cleaned and prepared for the machine learning module. We have to deal with both numerical and categorical features. For the numerical features we have to first find the outliers so that we have good data points and their distribution should not be skewed.Through plotting box plots we can easily visualize the distribution of the data in a particular column and can see the outliers. To actually find the exact outliers we have gone for IQR. It is a prey straight forward approach where in to find the upper limit we add 1.5*IQRto 75th percentile and for lower bound we subtract 1.5*IQR from 25th percentile.(table 2)

$$IQR = 75th\text{-}25th$$

$$UB = (1.5*IQR) + 75^{th}$$

$$LB = 25th - (1.5*IQR)$$

Column	Upper-bound	Lower-Bound
Production	16578.5	-9825.5
Area	10928.5	-6427.5

Anything outside the upper bound and lower bound is an outlier and should be removed.We can again plot the box plot to check if the distribution is even or not.

Normalizing

After removing the outliers we can still see that the Area and Production Columns have really large values when compared to other numerical columns.So it becomes necessary to normalize these values so that the machine learning model can predict beer. We have used a min-max scaler to normalize and scale the values before we go for the ML module. Min-Max scaler shrinks the data between 0 and 1 without changing the distribution of the data.

Mathematically min-max scaler can be seen as:

$$x\_std=(x–x(min))/(x(max)– x(min))$$

$$x\_scaled = x\_std * (max – min) + min$$

	Production	Area
Mean	0.089	0.107
Std	0.166	0.186
Min	0.000	0.000
Max	1.000	1.000

Encoding

Now after we have cleaned and normalized the numerical variables present in the dataset. We have to deal with categorical variables. Nominal variables don't have any order like we have in our dataset State Names and Crop Names. These two columns have to be encoded so that they can be taken as independent features in the Machine Learning Model.We have used exact encoding for both the columns. Effect/Sum encoding is like One hot encoding but instead of keeping some rows fully 0 it replaces these 0’s with -1 so that we don’t get a Sparse matrix. We can see after encoding both the columns we finally get 157 feature columns.

Evaluation

R2 Score: R2 score is the most common evaluation technique , or coefficient of determination that measures the proportion of the outcome's variation explained by the model, and is the default score function for regression methods.

k-fold cross validation: It is a great method for evaluating the performance of a model on a dataset. It is one of the most important methods used for this purpose. It basically divides the dataset into k almost equal pas or folds and then trains the model at hand on k-1 folds. It keeps one-fold aside for testing. It is repeated k times. Each time the MSE is calculated and at the end average MSE is given

Deployment

The deployment of the entire model has been done using a python library called Streamlit. It allows easy data optimization,deployment and statistical analysis with minimal amount of code.It also prevents any requirement of prior knowledge of deploying web service frameworks like Django and Flask. This can be extremely useful in building data dashboards especially when the team consists of mostly non-tech members.Streamlite is easy to use since it uses predefined commands to build an interactive data driven web application.Simple commands like st.write() to implement a wide variety of objects right from simple text to pandas dataframe matplotlib visualizations becomes possible.Our deployed model consists of a landing homepage along with the navigation sidebar that the user can interact with . The homepage consists of the basic information about the project and accustoms the user to the entire problem statement.The dataset page displays the project's dataset along with its description in a tabular form using the pandas data frame function .The graphs section consists of various interactive graphs plotted using the matplotlib libraries Lastly the model page comprises the user input parameters which include the crop name, state name , amount of area , temperature and rainfall for predicting the target production levels in ton. The user is also able to select the algorithm that they want to apply for beer comparison and accuracy after clicking on the run button. Given below is a pseudo code for the Streamlit data driven web application.

RESULTS AND DISCUSSION

Exploratory Data Analysis

After performing cleaning and normalizing the datasets we performed Exploratory Data Analysis to get a beer understanding of the agricultural scenario of India. Through plotting and analyzing the graph I came to know a lot of useful information and could understand the dataset beer.

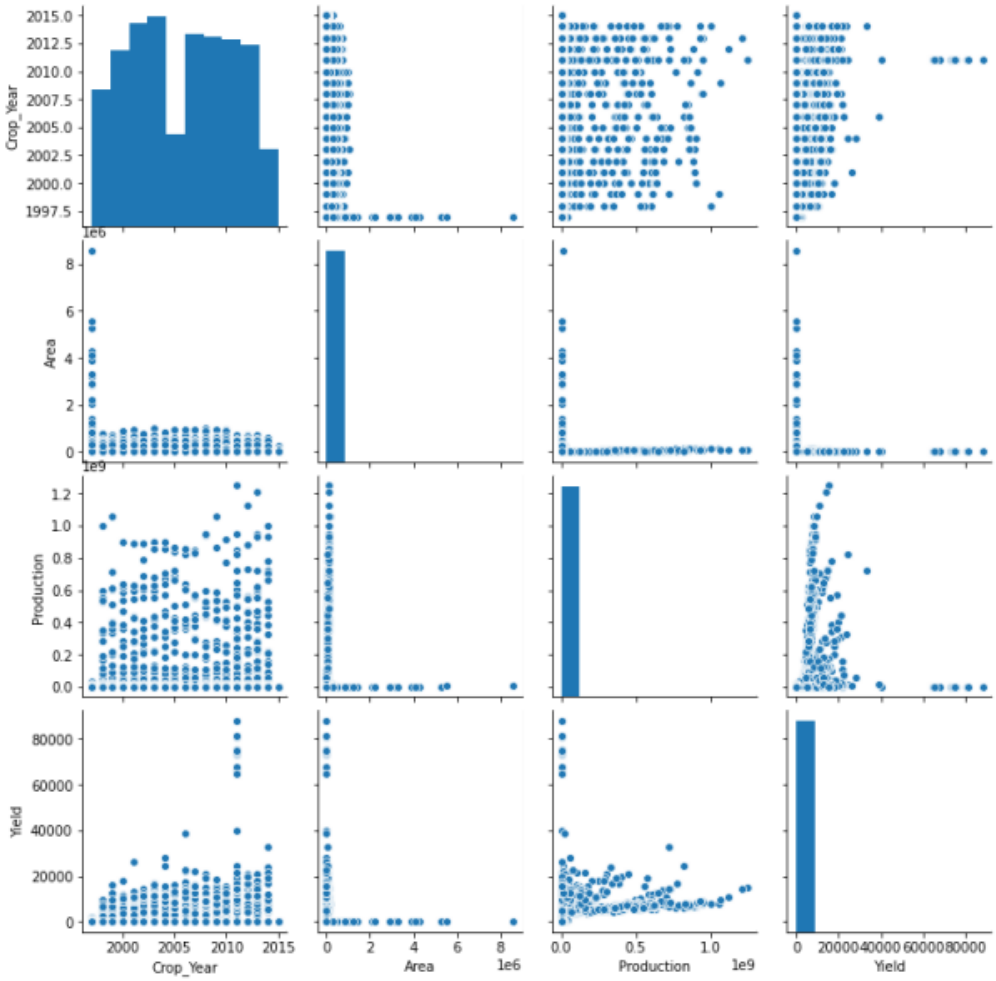


Figure 1. Crop-wise total production from 1997 - 2015

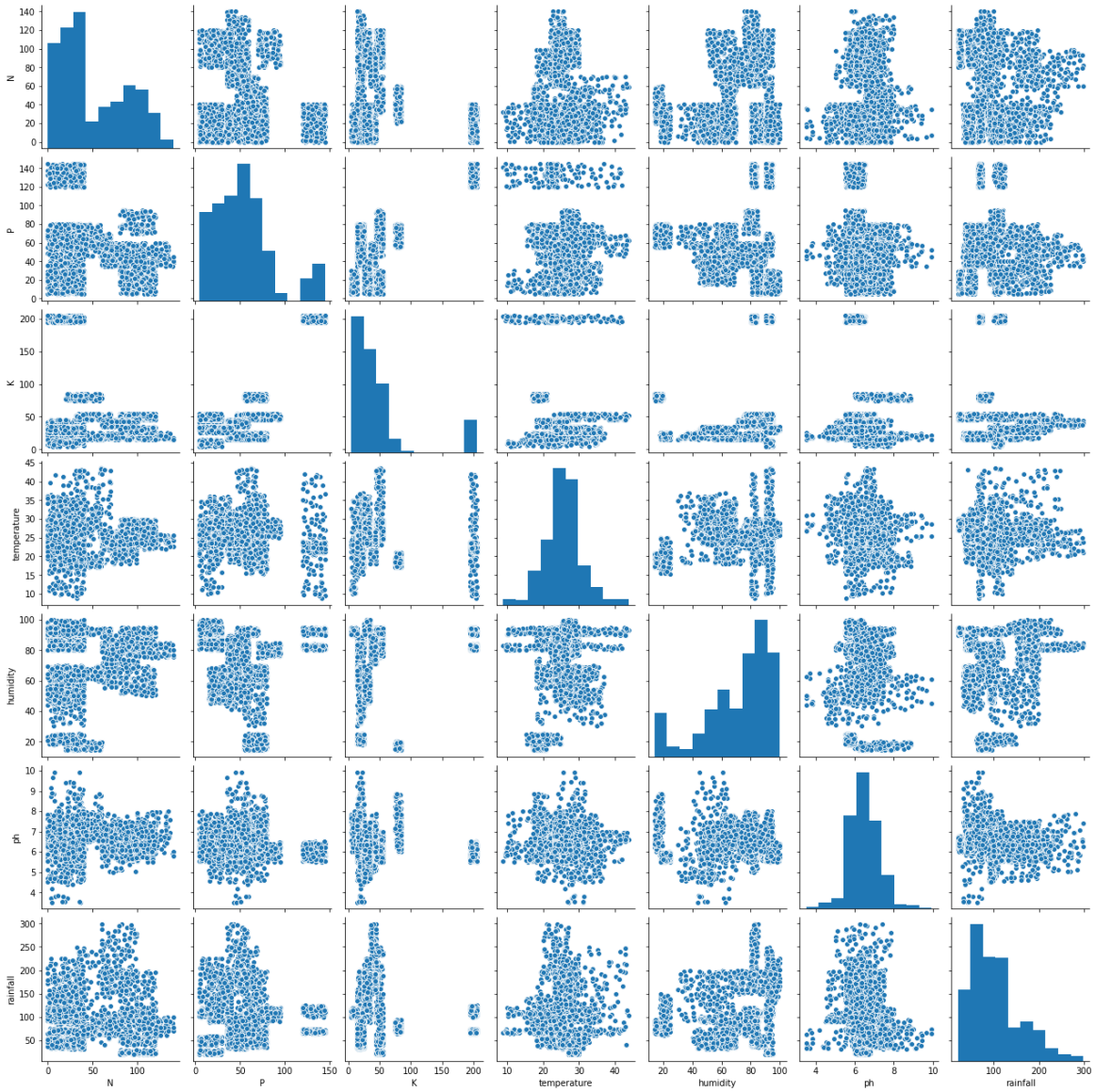


Figure 2. Comparison between rainfall, ph,humidity,temperature,N,P,K

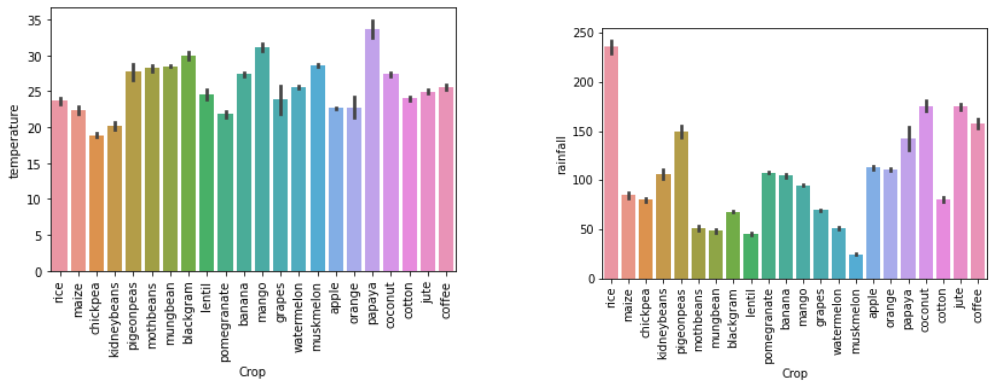


Figure 3. Comparison of crop production and average rainfall and comparison of crop production and average temperature

Figure 3 highlights the comparison of crop production with the average rainfall over the years. It is quite evident from the graph that with an increase in rainfall the productivity levels have subsequently shot up whereas whenever a drop of rainfall is

seen the productivity also falls. This graph clearly highlights the high dependency of the Indian crops on rain. Figure 13 deals with the comparison of average temperature with the total crop production over the years. The crop production levels have been inversely related to the temperature levels recorded with very high or very low temperature causing major dips in the production levels.

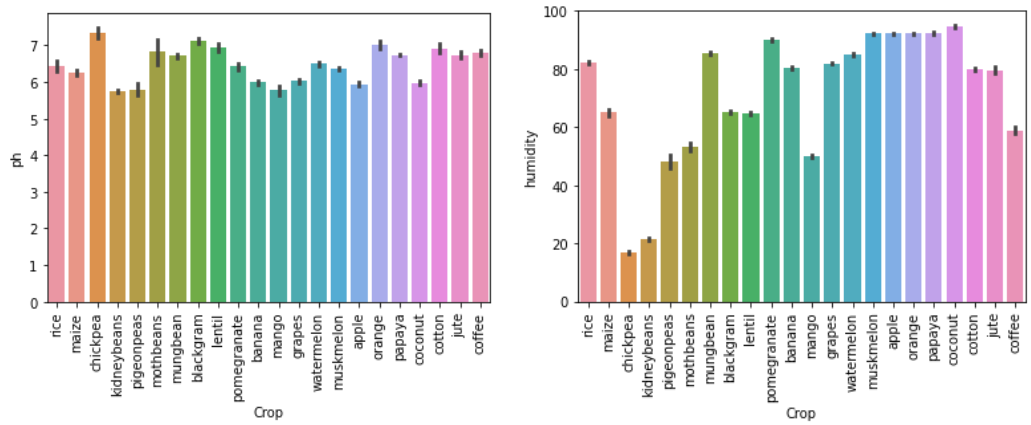


Figure 4. Comparison of crop production and ph and comparison of crop production and humidity

Figure 4 illustrates the correlation between crop production and soil pH levels throughout the observed years. The graph vividly depicts a discernible pattern wherein crop productivity experiences a notable surge with an increase in soil pH. Conversely, a decline in crop production is evident when there is a decrease in soil pH. This graphical representation underscores the significant impact of soil pH on crop yields, emphasizing the importance of maintaining optimal pH levels for sustained agricultural productivity. In a similar vein, Figure 4 delves into the comparison of crop production with humidity variations over the years. The graph underscores a clear relationship between humidity levels and crop productivity, showcasing that higher humidity is associated with enhanced crop yields. Conversely, a decrease in humidity corresponds to a decline in crop production. This graphical analysis accentuates the pronounced influence of humidity on the agricultural landscape, providing valuable insights for farmers and policymakers aiming to optimize crop production in varying environmental conditions.

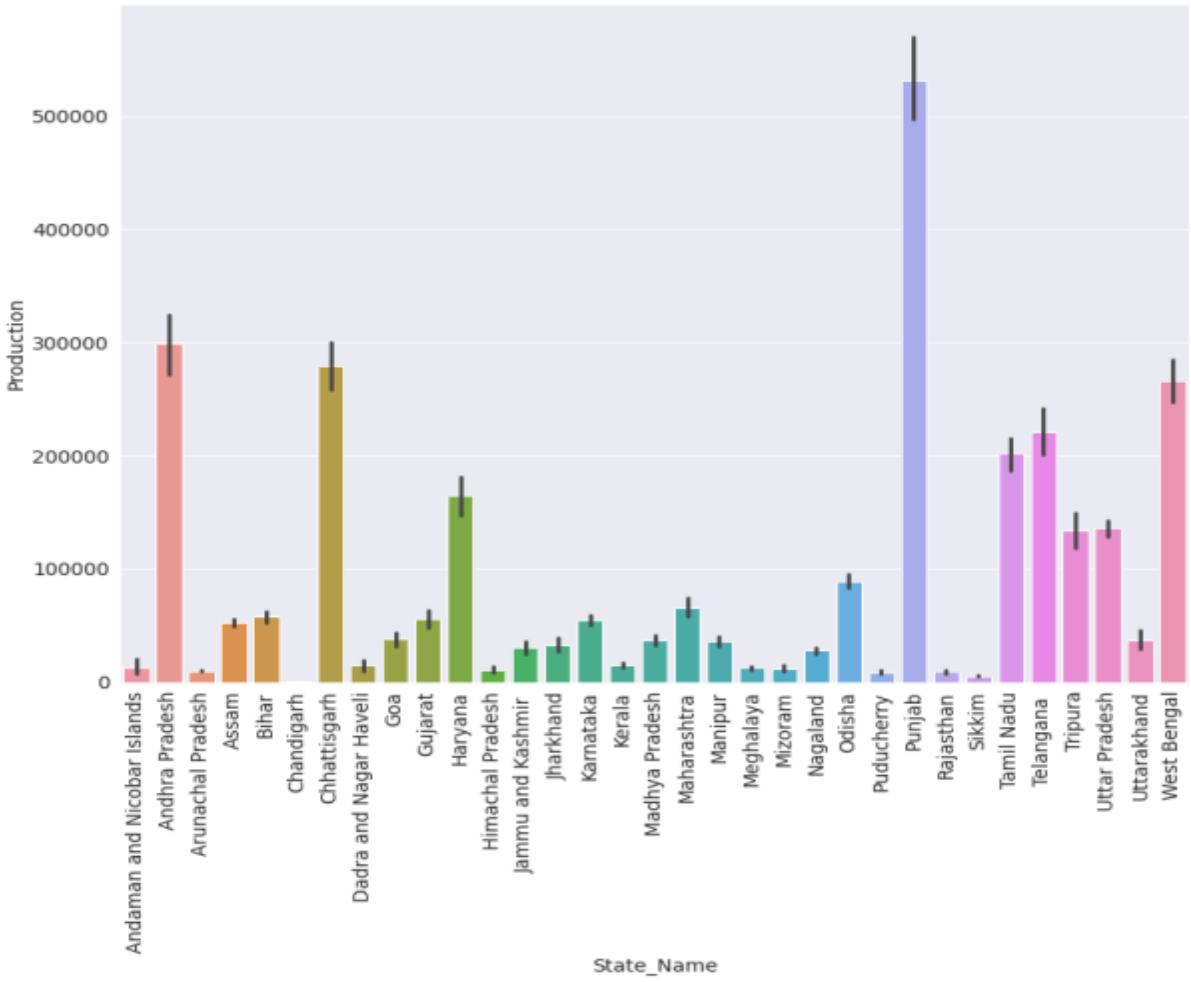


Figure 5. Production of Crop in each state

Figure 5 presents a comprehensive overview of crop production distribution across different states. The graph reveals distinctive patterns in agricultural output, with certain states emerging as key contributors to the nation's overall production. Notably, states with favorable climatic conditions and robust agricultural practices showcase higher crop yields, contributing significantly to the country's food production. Conversely, regions facing challenges such as water scarcity or adverse weather conditions exhibit fluctuations in crop production. This graphical representation serves as a valuable tool for policymakers and stakeholders, offering insights into regional disparities and highlighting areas where targeted interventions may be necessary to enhance overall agricultural productivity and ensure food security across diverse geographical landscapes.

Results

After our Exploratory Analysis of the dataset we arrived at various facts about the Indian agricultural scenario as mentioned in observation. We trained 4-5 Machine Learning Models on the nal dataset. We have calculated the R2 score and results are as follows

Performance of the Algorithms

- Naive Bayes Classification:

Accuracy of the Algorithm Naive Bayes Classification was 0.9436363636363636
Given below is the confusion matrix of Naive Bayes Classification algorithm.

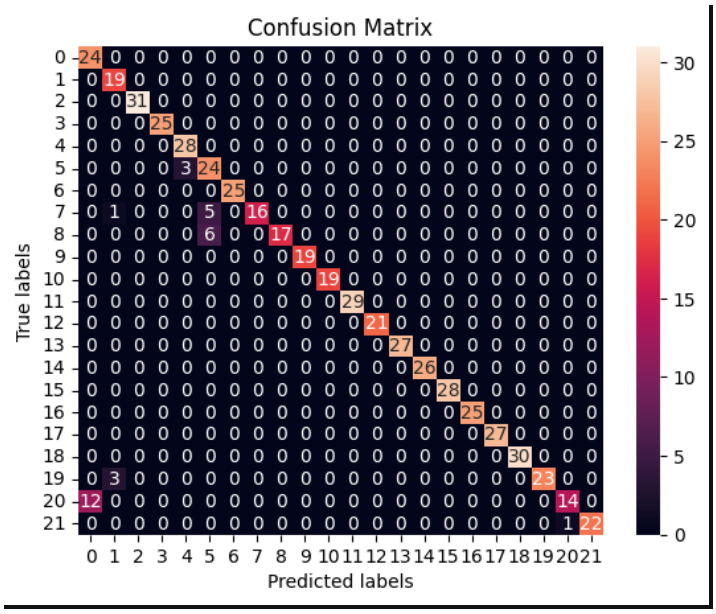


Figure 6. Confusion Matrix of Naive Bayes Classification

- Decision Tree Classification:

Accuracy of the Algorithm Decision Tree Classification was 0.9563636363636364 .
Given below is the confusion matrix of Decision Tree Classification algorithm.

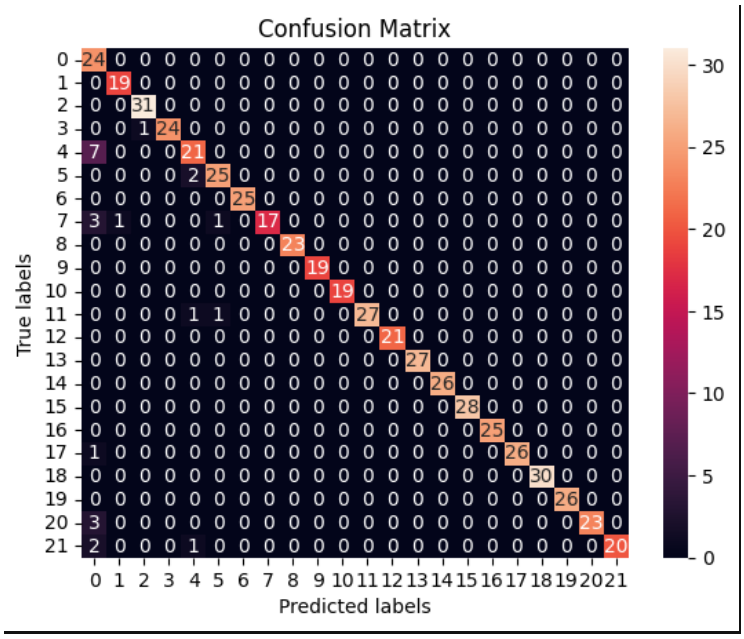


Figure 6. Confusion Matrix of Decision Tree Classification

- Random Forest Classification:

Accuracy of the Algorithm Random Forest Classification was 0.9818181818181818 . Given below is the confusion matrix of Random Forest Classification algorithm.

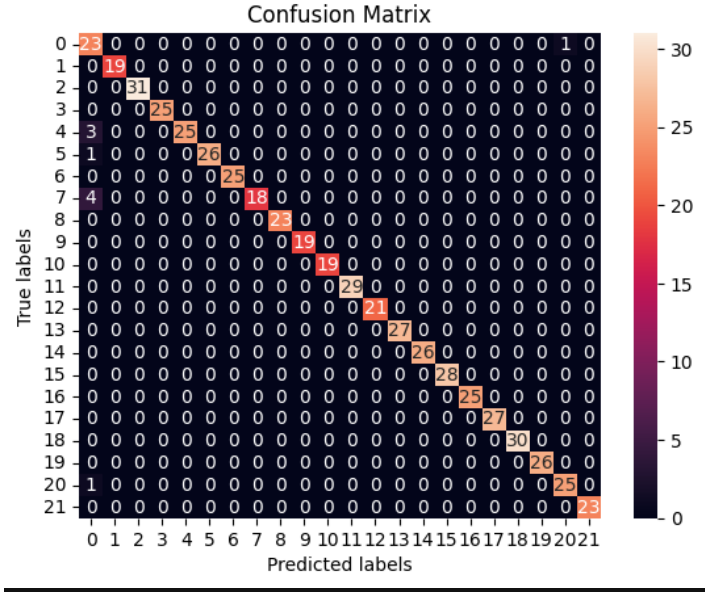


Figure 6. Confusion Matrix of Random Forest Classification

- KNN Classier:

Accuracy of the Algorithm KNN Classier was 0.9745454545454545. Given below is the confusion matrix of KNN Classier algorithm.

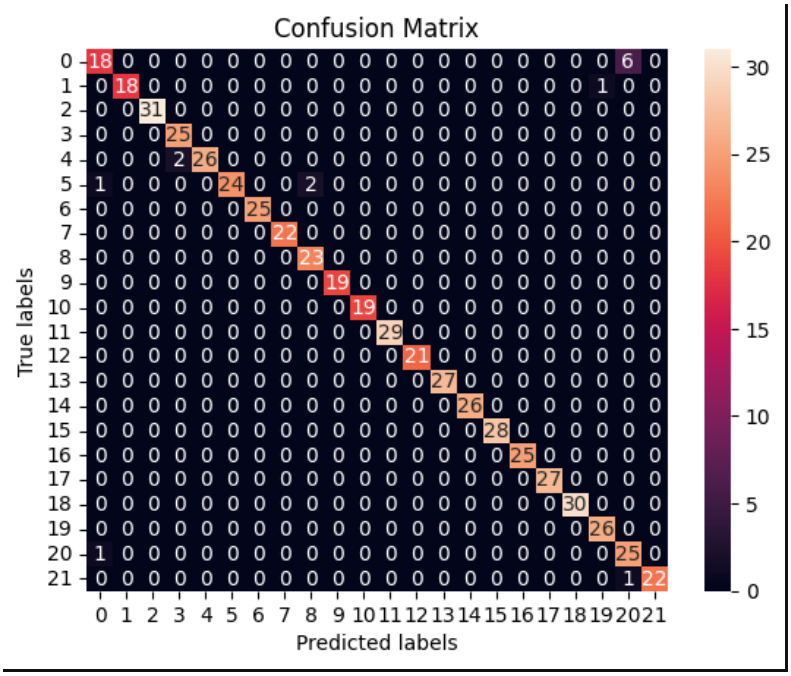


Figure 6. Confusion Matrix of KNN Classier

- Gradient Boosting:

Accuracy of the Algorithm Gradient Boosting was 0.9818181818181818 . Given below is the confusion matrix of Gradient Boosting algorithm.

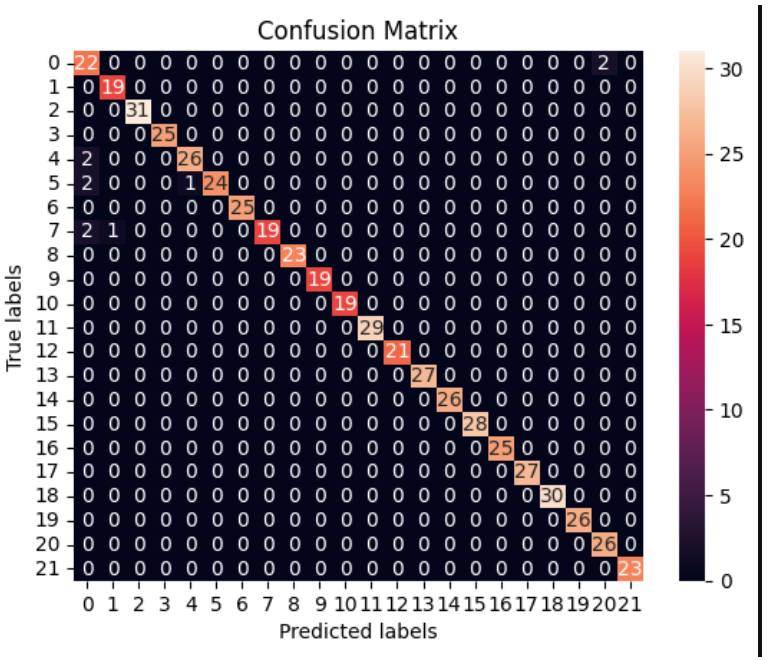
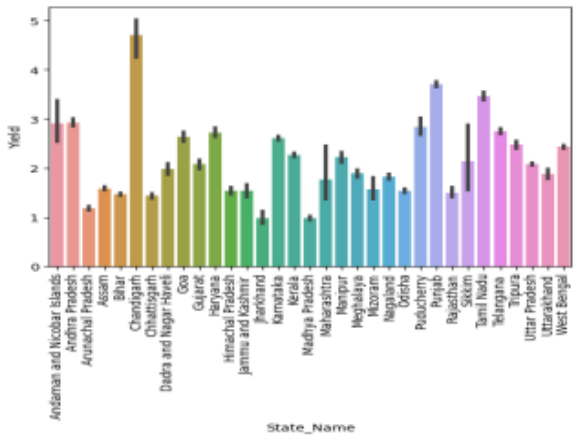


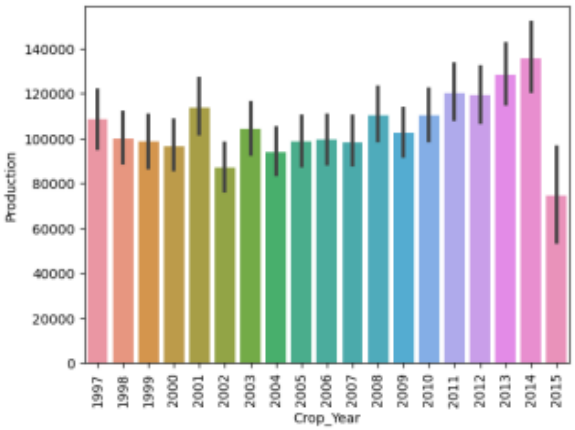
Figure 6. Confusion Matrix of Gradient Boosting

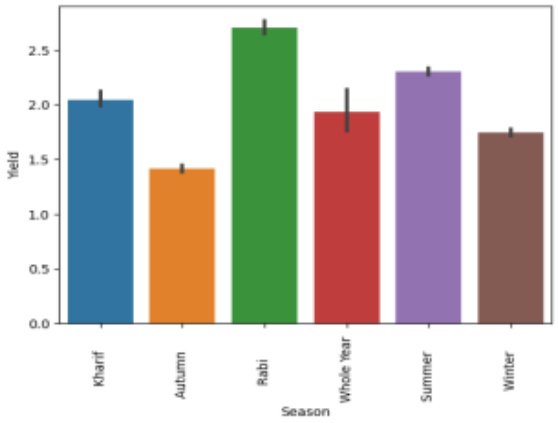
Visualization

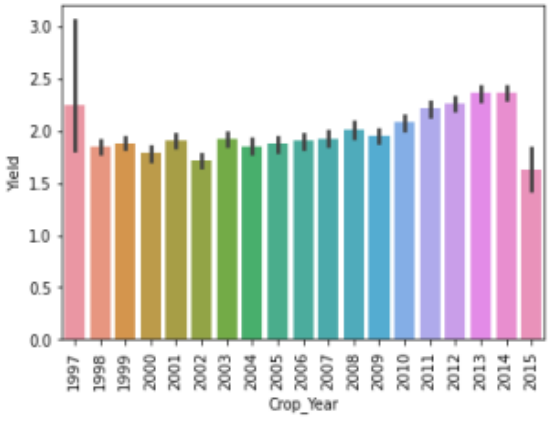
Analyzing each type of Crop

1. Rice:



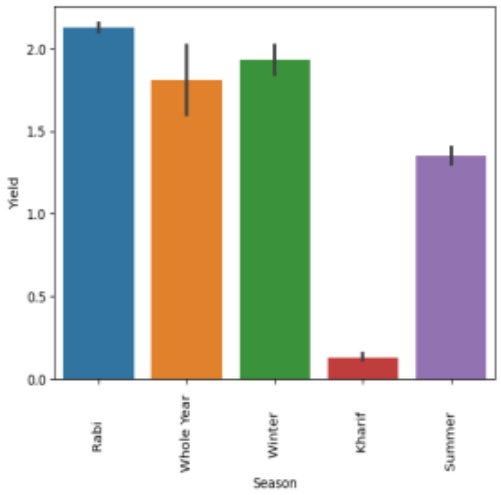


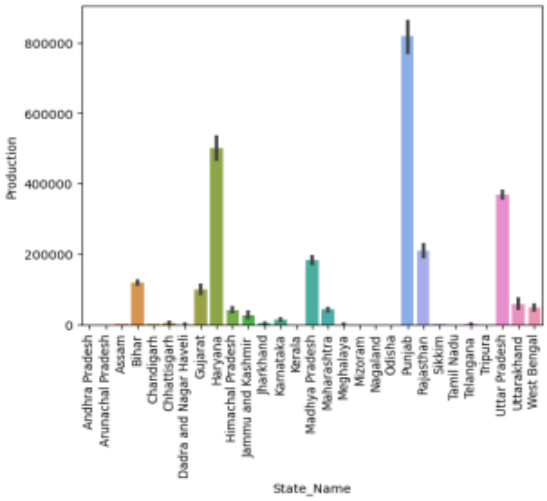


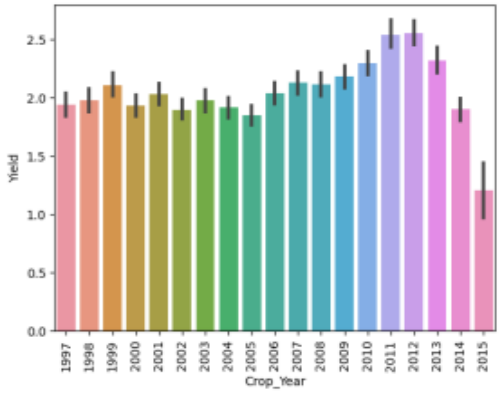


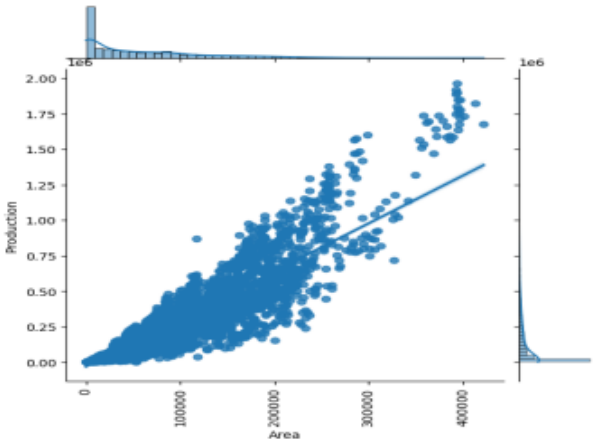
Observations Obtained:
1. Rice yield is maximum in Rabi season.
2. 2. Rice yield is maximum in Chandigarh.
3. Rice yield has been growing a little from the year 2009 to 2014.

2. Wheat:





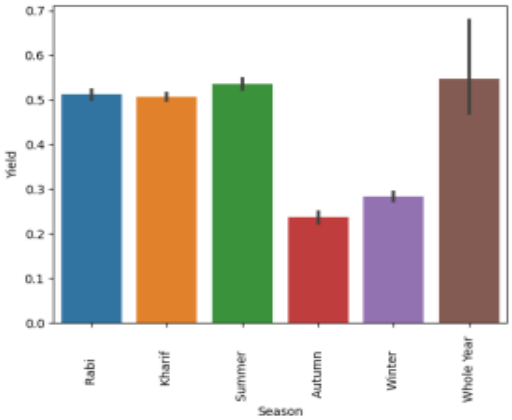


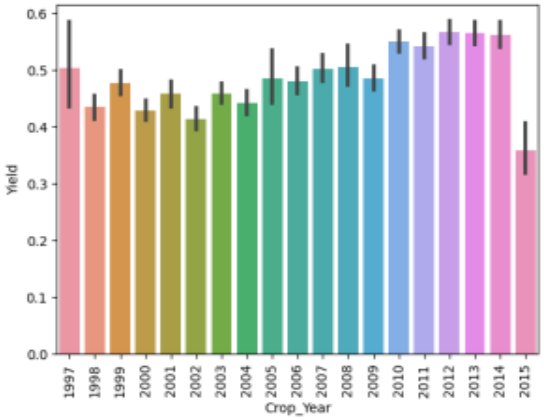


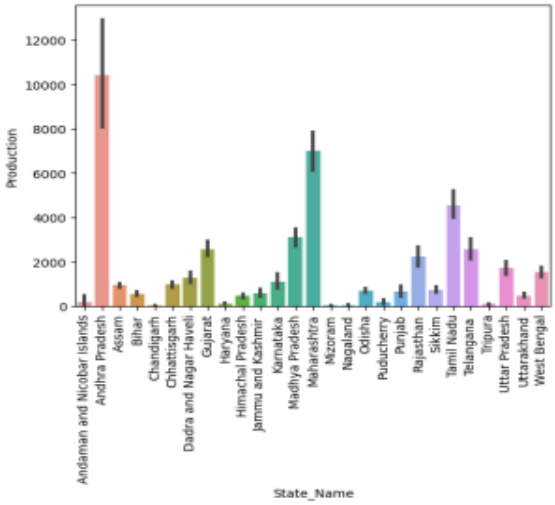
Observations Obtained :

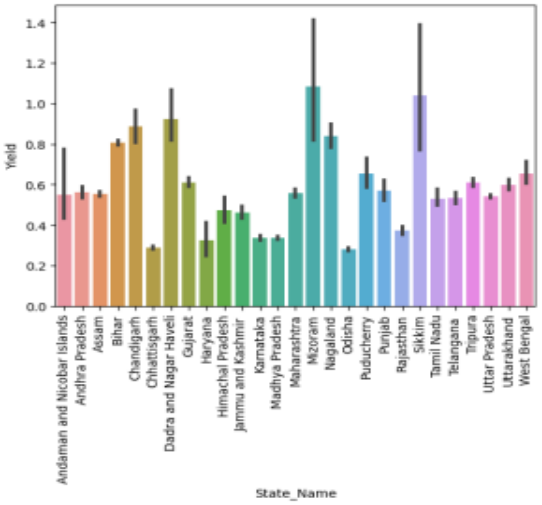
1. Wheat yield is maximum in Rabi season.
2. Wheat yield is maximum in Punjab.
3. Coconut yield is decreasing in the year 2012 to 2015

3.Coconut:





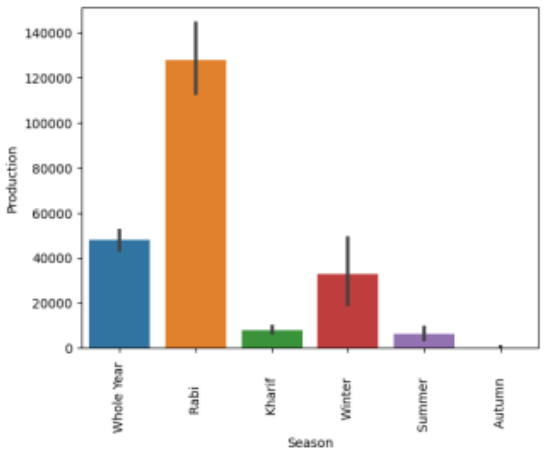


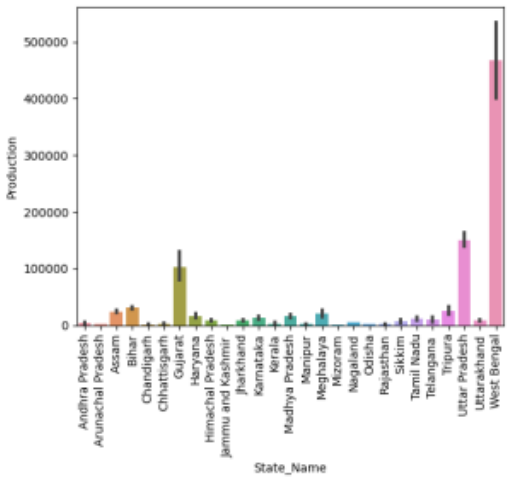


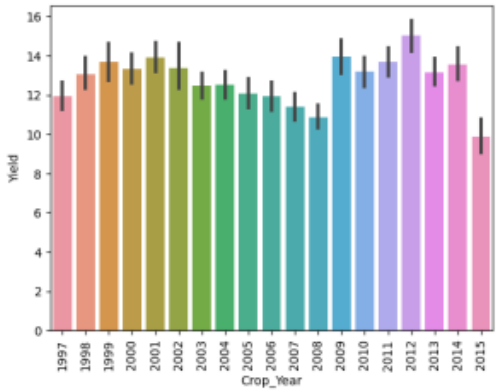
Observations Obtained :

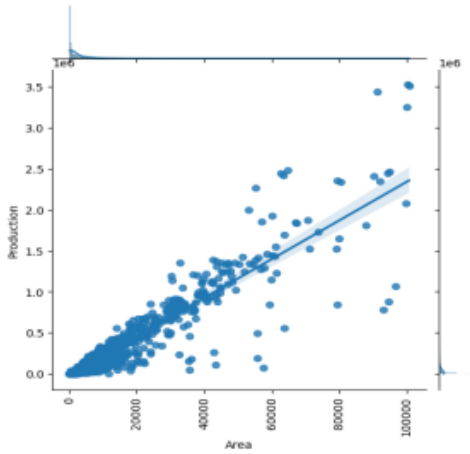
1. Andhra Pradesh is the largest producing coconut state.
2. Production per unit area is higher in Mizoram and Sikkim.
3. Coconut yield is decreasing in the year 2012 to 2015.

4. Potato:





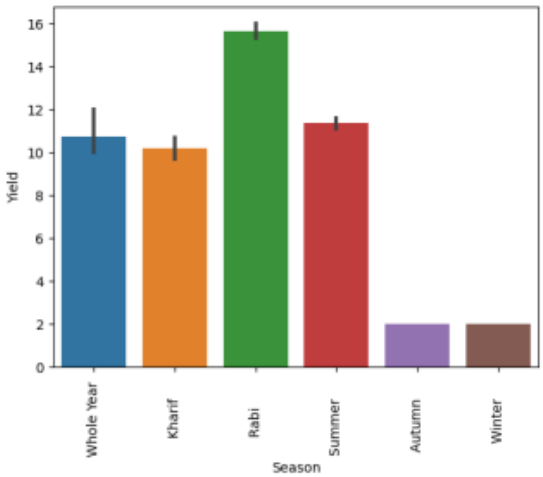


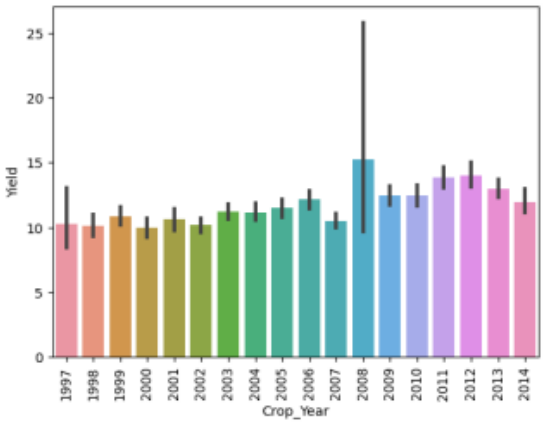


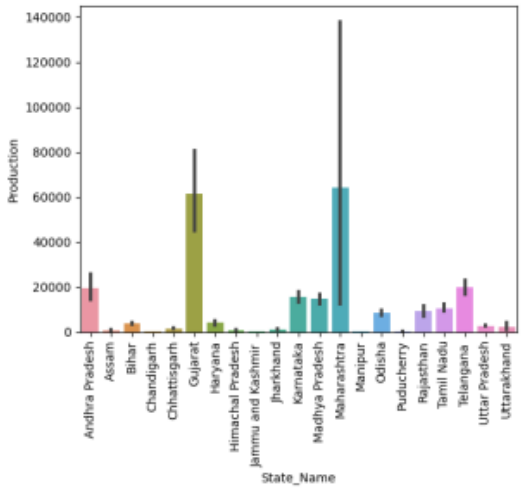
Conclusions Obtained :

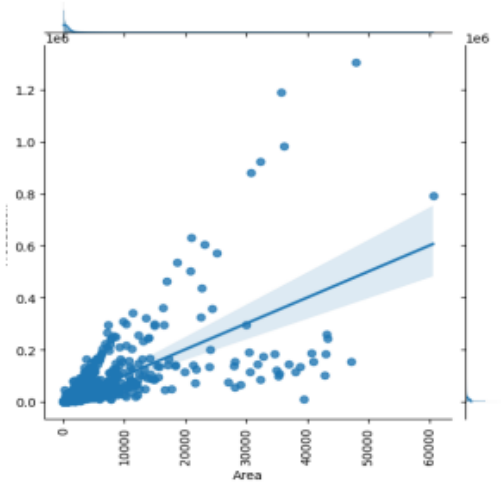
1. Potatoes are a Rabi crop.
2. West Bengal is the largest producer of potatoes.
3. Potato yield is decreasing in the year 2001 to 2008.

5.Onion:





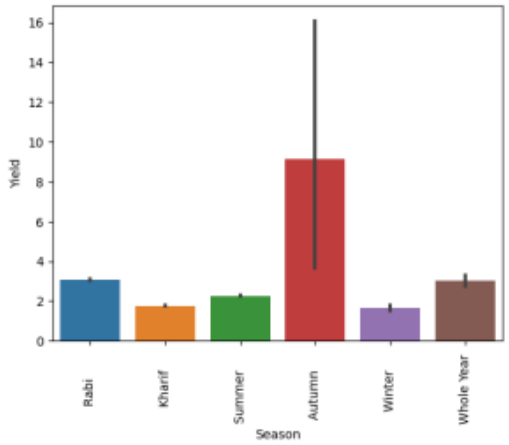


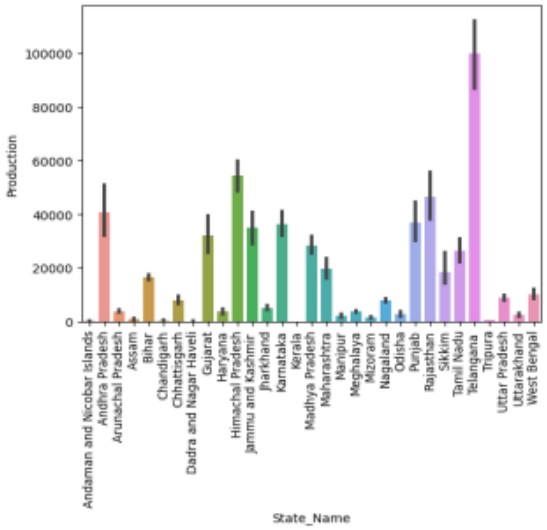


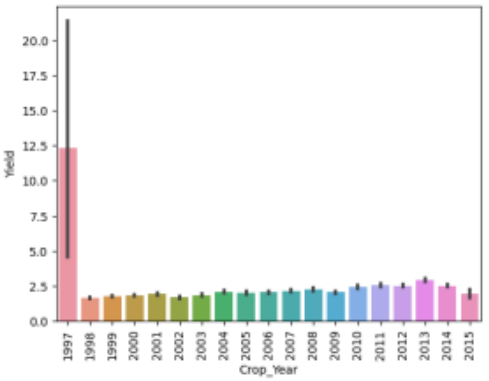
Observations Obtained:

1. Onion is a Rabi crop.
2. Gujarat and Maharashtra are the major onion-producing states.
3. Onion yield is decreasing in the year 2012 to 2014.

6.Maize:







Observations Obtained:

1. Maize is produced in the autumn season
2. Telangana is the major maize-producing state.
3. There was a sudden decline in maize production from the year 2000.

Conclusion

We can finally conclude the following about the project: Through Exploratory data analysis we could see that Indian crops are directly related to rainfall and negatively to temperature. We could also conclude that the kharif season contributes most to the crop production. The Machine Learning Algorithms Gradient Boost and Random forest Classification had the highest accuracy of about 84% and 84%-76% respectively. Ridge Regression and Gradient Boosting Regression performed fairly well with accuracy of about 54% and 66% respectively. The Data Driven Web application was able to display all the information regarding the project. All the Machine Learning algorithms were successfully deployed on the application. This project signifies the importance of data science and is easy to use in the modern world. The project does come a bit under pressure when compared to specific crop-based applications. It also takes a lot of time to train the complex ML models on this large dataset and thus will require more powerful computers to run it. The proposed Crop Yield Prediction Algorithm (CYPA) incorporates multiple data sources such as climate, weather, agricultural yield, and chemical data to anticipate annual crop yields by policymakers and farmers in their country. The study demonstrated the efficacy of CYPA by training and verifying five models using optimal hyper-parameter settings for each machine learning technique. The results indicate that CYPA can achieve high accuracy in predicting crop yields, as demonstrated by the scores of Decision Tree Classier, Random Forest Classier, and KNN Classier. Additionally, the paper introduces a new algorithm based on active learning that can enhance CYPA's performance by reducing the number of labeled data needed for training. Incorporating active learning into CYPA can improve the efficiency and accuracy of crop yield prediction, thereby enhancing decision-making at international, regional, and local levels. Overall, this study highlights the potential of IoT and machine learning techniques in addressing the critical challenge of predicting crop yields, thereby facilitating informed decision-making for policymakers and farmers alike. When utilizing Decision Tree Classier, Random Forest Classier, and KNN Classier, the score is equal to 0.95, 0.98, and 0.97, respectively.

Future Work

This data is ever growing and will require regular updating so that the model is up-to-date. Since this is a huge dataset it would require powerful computers to run more complex algorithms like SVM etc. These algorithms might work beer on the dataset and may give beer and accurate results. Deep Learning can also be used here in the future to improve the accuracy to a greater extent. This analysis is just a tip of iceberg, with nineteen year crop production data, a lot could be done and some of the ideas are:

- Instead of deleting missing data for Production(3730 data points), we could impute based on the area used for cultivation and state.
- Zone wise cultivation status and predict future production prediction using regression.
- Crop Categories and status of their cultivation over the years, if the production has gone up (Good case scenario) and if production has gone down (bad case scenario)…can we look into the causation of this trend.
- Asking further important questions like, Kerala is low in area coverage compared to other southern states but still in production levels it's high why?

References and Bibliography

- https://en.wikipedia.org/wiki/Green_Revolution_in_India
- https://www.kaggle.com/datasets/abhinand05/crop-production-in-india

- Abdul Razzaq Ghumman, Ateeq-ur-Rauf, Husnain Haider and Md. Shaquzamman , “Functional data analysis of models for predicting temperature and precipitation under climate change scenarios”, Journal of Water and Climate Change,2019.

- Bhadouria R, Singh R, Singh VK, Bohakur A, Ahamad A, Kumar G, Singh P (2019) Chapter 1 - agriculture in the era of climate change: consequences and effects. In: Choudhary KK, Kumar A, Singh AK (eds) Climate change and agricultural ecosystems. Woodhead Publishing, Sawston, pp 1–23

- Xu X, Gao P, Zhu X, Guo W, Ding J, Li C, Zhu M, Wu X (2019) Design of an integrated climatic assessment indicator (ICAI) for wheat production: a case study in Jiangsu Province, China. Ecol Indic 101:943–953.

- Alpaydin E (2010) Introduction to machine learning, 2nd edn. MIT Press, Cambridge.