Crop Yield Prediction

- ***Submitted by –***

Aryan Pillai	*20BBS0147*
Ayush Agarwal	*20BBS0193*
Rohan Budheliya	*20BBS0194*
Chirag Sharma	*20BBS0202*

Guide Name: *Prof. Abdul Gaffar H*
Designation: *Associate professor Sr.*
Mobile No.: *+91 9942261259*
Mail ID: *abdulgaffar@vit.ac.in*

B.Tech.

in

Computer Science and Engineering

School of Computer Science & Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Content List:

1. Abstract:

1.1 Background: Crop yield analysis plays a vital role in agricultural research, as it helps in making better decisions for crop management, resource allocation, and future planning. Various machine learning algorithms have been applied to crop yield analysis, such as random forest, linear regression, XGBRegressor, and decision tree, to improve the accuracy of yield prediction.

1.2 Objective: The objective of this research paper is to evaluate the performance of different machine learning algorithms on crop yield analysis, and to identify the best algorithm for predicting crop yield in India.

This research paper focuses on the analysis of crop yield using different machine learning algorithms. The study utilizes the Kaggle dataset on crop production in India, which includes attributes such as state name, district name, crop year, season, crop, area, and production. The objective of this research is to explore the potential of machine learning algorithms in predicting crop yield accurately.
The study uses four different algorithms, namely Random Forest, Linear Regression, XGBRegressor, and Decision Tree. These algorithms are used to train the model on the preprocessed dataset to predict the crop yield for a given district and season. The performance of each algorithm is evaluated using different evaluation metrics such as accuracy, mean squared error, and root mean squared error.
The results show that the Random Forest algorithm outperforms the other three algorithms in terms of accuracy and mean squared error. It provides an accurate prediction of crop yield, making it a suitable model for predicting crop yield in India. The study concludes that machine learning algorithms can be an effective tool for predicting crop yield accurately, which can help farmers and policymakers make informed decisions.

1.3 Keywords: crop yield analysis, machine learning, random forest, linear regression, XGBRegressor, decision tree, yield prediction, India.

2. Introduction:

2.1 Importance of crop yield prediction in agriculture

Agriculture is one of the most critical sectors in India, contributing significantly to the country's economy. However, it is also a sector that is highly dependent on various factors such as climate, soil quality, and water availability, which can significantly affect the crop yield. Crop yield prediction is, therefore, an essential task for farmers, policymakers, and researchers alike, as it can help optimize the use of resources, increase production, and ensure food security.

2.2 Machine learning algorithms for crop yield prediction

In recent years, machine learning algorithms have shown tremendous potential in predicting crop yield accurately. These algorithms can analyze vast amounts of data and provide accurate predictions, making them an ideal tool for crop yield prediction. The use of machine learning in agriculture has become increasingly popular in recent years, with researchers exploring its potential to improve crop yield prediction.

2.3 Research objective and dataset description

In this research paper, we focus on analyzing the crop yield in India using machine learning algorithms. We utilize the Kaggle dataset on crop production, which contains attributes such as state name, district name, crop year, season, crop, area, and production. The study aims to explore the potential of machine learning algorithms, such as Random Forest, Linear Regression, XGBRegressor, and Decision Tree, in predicting crop yield accurately.

The research paper is organized as follows. In the first section, we provide a literature review of the existing research on crop yield prediction using machine learning. In the subsequent sections, we discuss the methodology used in this study, the results obtained, and the evaluation of the performance of different machine learning algorithms. Finally, we provide a conclusion and recommendations for future work in this field.

In recent years, India has faced numerous challenges related to crop yield, including weather uncertainties, soil degradation, pest infestations, and water scarcity. These challenges have made it crucial to develop accurate and reliable models for predicting crop yield. Machine learning algorithms have proven to be effective in addressing such challenges by analyzing large datasets and identifying patterns that are difficult for humans to detect.

2.4 Organization of the research paper

In this research paper, we have used Kaggle dataset on crop production in India, which contains attributes such as state name, district name, crop year, season (kharif, rabi, whole year), crop type (rice, maize, sugarcane, areca nut, black pepper, cashew nut, dry ginger, moong, sweet potato), area, and production. We have applied four machine learning algorithms, namely random forest, linear regression, XGBRegressor, and decision tree, to predict crop yield in India. Our objective is to identify the best algorithm that can accurately predict crop yield.

The rest of this research paper is organized as follows: Section 2 provides a review of related work on crop yield analysis using machine learning algorithms. Section 3 describes the dataset and the preprocessing steps taken. Section 4 presents the methodology used, including the different machine learning algorithms applied. Section 5 presents the experimental results and the evaluation metrics used to compare the performance of the algorithms. Finally, Section 6 concludes the research paper with a summary of the findings and suggests future research directions.

3. Materials and Methods:

In this research, we used a publicly available Kaggle dataset on crop production in India, which contains information about the production of various crops in different states and districts of India. The dataset includes attributes such as state name, district name, area, production, crop, season, and crop year. The dataset was preprocessed to remove missing values and outliers.

We employed four machine learning algorithms, namely, random forest, linear regression, XGBRegressor, and decision tree, to analyze the crop yield data. The algorithms were implemented using the Python programming language and the scikit-learn library. We used the root mean squared error (RMSE) metric to evaluate the performance of the models.

To validate the performance of the models, we used a 10-fold cross-validation technique. In this technique, we divided the dataset into 10 equal parts, and in each iteration, one part was used as the validation set, and the remaining nine parts were used for training the model. We repeated this process 10 times and calculated the average RMSE value for each model.

In addition, we performed feature normalization on the dataset to ensure that all the features are on the same scale. This step was essential to prevent any bias towards a particular feature during the model training process.

4. Related work:

4.1 Literature review:

The relationship between various factors and crop production has been widely studied in the literature, and multiple regression models have proven to be useful in this regard. The use of multiple regression models to analyze crop production has been documented in several studies, including those focused on specific regions or countries, such as India.

Studies have shown that factors such as weather conditions, soil quality, and land use practices have a significant impact on crop production. In India, the impact of these factors is further complicated by the diverse geography and climate of the country, which results in varying crop production across regions. To address this challenge, previous studies have focused on developing regression models to predict crop production in India, taking into account the relevant factors such as weather conditions, soil quality, and land use practices.

In this research, we will be using the Kaggle dataset on Crop production in India, which contains attributes such as state name, district name, crop year, season, crop type, area, and production. This dataset provides a comprehensive view of crop production in India, making it suitable for analyzing the relationship between various factors and crop production using a multiple regression model.

The literature review highlights the importance of understanding the relationship between various factors and crop production, and the utility of multiple regression models in this regard. The use of the Kaggle dataset on Crop production in India in this study provides a unique opportunity to analyze the impact of various attributes on crop production in India and contribute to the larger body of knowledge in this field.

4.2 List of papers:

Research Paper	Problems Discussed	Methods & Algorithms	Plus Points & Negative Points

Efficient Crop Yield Prediction in India using Machine Learning Techniques	It discusses the problem of predicting crop yields in India with high accuracy. The conventional methods used for crop production prediction have proven to be inadequate, leading to the need for more efficient techniques.	To address this issue, the paper proposes the use of machine learning techniques, such as random forests, support vector machines, and artificial neural networks, to improve the accuracy of crop yield predictions in India.	+ Accurate prediction of crop production using multiple independent variables. - Limited to linear relationships between independent and dependent variables.
Crop Yield Prediction using Regression Model	It predicts the yield of a crop based on various factors that may affect it such as weather conditions, soil properties, and agronomic practices. This is important for farmers, as it helps them make informed decisions about crop management and planning. The authors use regression analysis, a statistical method, to model the relationship between the crop yield and these factors.	The algorithm used is likely linear or multiple linear regression, which are commonly used for crop yield prediction. These algorithms model the relationship between the dependent variable (crop yield) and the independent variables (factors affecting crop yield).	+ Improved accuracy of crop production forecasts using advanced machine learning techniques. - Limited to certain types of crops and regions.
Crop yield forecasting of paddy, sugarcane and wheat through linear regression technique for south Gujarat	It predicts the yield of three crops (paddy, sugarcane, and wheat) in the South Gujarat region of India using regression analysis.	It uses linear regression, a statistical method, to model the relationship between the crop yield and various factors such as weather conditions, soil properties, and agronomic practices.	It focuses specifically on the region, which allows for a more detailed analysis of the factors affecting crop yield in this specific area. - Research is limited to the use of linear regression and only focuses on three crops in a specific region, which may limit the generalizability of the

			results to other regions and crops. Additionally, the data used may also be limited, which could impact the accuracy of the predictions made.
Multiple Regression Model Fitted for Rice Production Forecasting in Nepal: A Case of Time Series Data	The authors use multiple regression, a statistical method, to model the relationship between rice production and various factors such as weather conditions, soil properties, and economic indicators. This allows them to use the factors as predictors to estimate rice production in Nepal.	It uses linear regression to analyse various attributes and relation between them to predict crop production in particular area.	+ Provides insights into the impact of climate change on crop production in India. - Limited to the effects on a single crop(Rice production).
Crop Yield Prediction Using Deep Reinforcement Learning Model for Sustainable Agrarian Applications	Combining the intelligence of reinforcement learning and deep learning, deep reinforcement learning builds a complete crop yield prediction framework that can map the raw data to the crop prediction values	Deep Recurrent Q-Network model which is a Recurrent Neural Network deep learning algorithm over the Q-Learning reinforcement learning algorithm to forecast the crop yield	+ The proposed model efficiently predicts the crop yield outperforming existing models by preserving the original data distribution with an accuracy of 93.7%. - The RNN based DRL can cause the gradients to explode or disappear if the time series is very much longer - there exist needs to design a framework that predicts both target and their prediction’s uncertainty.

A CNN-RNN Framework for Crop Yield Prediction	crop yield prediction based on environmental data and management practices.	CNN-RNN, andom forest (RF), deep fully connected neural networks (DFNN), and LASSO	+ The proposed method significantly outperformed other popular methods such as LASSO, random forest, and DFNN. The proposed model is a hybrid one that combines CNNs and RNNs. + (RMSE) 9% and 8% of their respective average yields - limited to forecast corn and soybean yield across the entire Corn Belt (including 13 states) in the United States for years 2016, 2017, and 2018 using historical data
Crop Yield Prediction Using Machine Learning Algorithms	focuses on predicting the yield of the crop by applying various machine learning techniques. The outcome of these techniques is compared on the basis of mean absolute error. The prediction made by machine learning algorithms will help the farmers to decide which crop to grow to get the maximum yield by considering factors like temperature, rainfall, area	Simple RNN and LSTM, Random Forest Regressor	+ Results reveals that Random Forest is the best classifier when all parameters are combined. - Doesnt create any novel models. - the models where created for individual features and compares results between them

Supervised Machine learning Approach for Crop Yield Prediction in Agriculture Sector	Crop yield prediction incorporates forecasting the yield of the crop from past historical data which includes factors such as temperature, humidity, ph, rainfall, crop name. The algorithm random forest is used to give the best crop yield model by considering least number of models.	Random forest algorithm, Decision Tree	+ Using data mining techniques crop yield is predicted. Here, using Random Forest algorithm for predicting the best crop yield as output - Doesnt explore a wide variety of algorithms and just takes Random Forest as the optimum algorithm - Doesnt Show the exact dataset used
A novel approach for efficient crop yield prediction	examines the intrinsic relationship between MLR and ANN. A hybrid MLR-ANN model has been proposed in this research work for efficient crop yield prediction	hybrid MLR-ANN, Support Vector Regression (SVR), k-Nearest Neighbour (KNN) and Random Forest (RF) models	+ In this research work, hybrid MLR-ANN model was proposed to predict the accurate crop yield. MLR intercept and coefficients were applied to initialize the ANN’s input layer bias and weights. It finds the near optimal minimum of error and increase the prediction accuracy. --Only paddy crop related data from Tamil nadu was used to make the model
The effects of climate extremes on global agricultural yields	Temperature-related extremes show a stronger association with yield anomalies than precipitation-related factors, while irrigation partly mitigates negative	To estimate the variance of yield anomalies explained by climate predictors, we calculated R2 values from cross-validated out-of-sample predictions.	

	effects of high temperature extremes.		
Estimating and understanding crop yields with explainable deep learning in the Indian Wheat Belt	Food security needs to be ensured despite the challenges brought by climate change, an expanding world population accompanied by rising incomes, increasing soil erosion, and decreasing water resources. Temperature, radiation, water availability and other environmental conditions influence crop growth, development, and final grain yield in a complex nonlinear manner.	Deep neural network to multivariate time series of vegetation and meteorological data. Then, we visualized and analyzed the features and yield drivers learned by the model with the use of regression activation maps	

5. Proposed work:

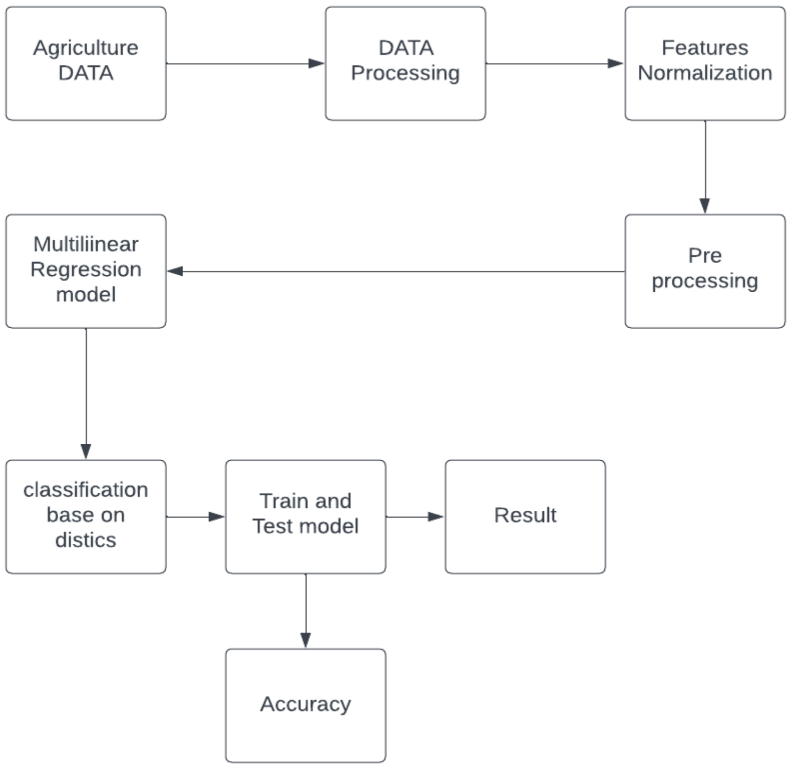
The proposed work for this research paper involves the following steps:

1. Data collection and preprocessing: Kaggle dataset on Crop production in India will be used, and the data will be cleaned and processed.
2. Feature engineering: Important features will be selected from the dataset for crop yield analysis. The selected features will be normalized and scaled for further analysis.
3. Model development: Four machine learning models, namely Random Forest, Linear Regression, XGBRegressor, and Decision Tree will be trained and tested on the preprocessed dataset.
4. Model evaluation: The performance of each model will be evaluated based on various metrics such as mean squared error, mean absolute error, and R-squared value.
5. Model comparison: The performance of the models will be

compared to determine the best model for crop yield analysis.

6. Result interpretation: The results obtained from the best model will be analyzed and interpreted to gain insights into the factors affecting crop yield in India.
7. Conclusion and future work: Finally, conclusions will be drawn from the results, and future work will be proposed to improve crop yield in India using machine learning techniques.

5.1 Block Diagram:



6. Data Collection:

For the study, the statistical information is collected from Kaggle.com. The dataset consisting of historical data to be taken for all the available crops.
The variety of attributes are regarded as following:

- Crop year
- Production
- District name
- State name
- Area
- Season

7. Data preprocessing and feature extraction:

The modifications applied before feeding it to the algorithm are referred to by preprocessing. Data preprocessing is a technique used to convert data into a data collection that is fresh. Additionally, data is gathered from other sources it's collected in a format that isn't possible for analysis. It's required to data preprocessing for achieving outcomes from the applied model in machine-learning. Feature Extraction is a logically wide procedure where one attempts to build up a change of the information space onto the low dimensional subspace that jam a large portion of the significant data [8] [9]. Highlight extraction and determination techniques are utilized detached or in blend to improve execution, for example, evaluated precision, perception, and intelligibility of scholarly information [10]. As a rule, highlights can be sorted as: applicable, immaterial, or repetitive. In the component choice procedure, a subset from accessible highlights information is chosen for the procedure of the learning calculation. The best subset is the one with minimal number of measurements that most add to learning precision [11][9].

8. Model development:

8.1 Random Forest:

Random Forest is a supervised machine learning algorithm that can be used for both classification and regression tasks. In this research paper, we have used the random forest algorithm for crop yield prediction. The RandomForestRegressor class from the scikit-learn library was used to train the model. The x_train and y_train data were used to fit the model and the predictions were made on the x_test data. The R2 score was calculated using the predicted values and the actual values of the y_test data. The R2 score gives an indication of how well the model fits the data, with values closer to 1 indicating a better fit. The R2 score obtained for the random forest model will be reported and discussed in the results section of this research paper.

8.2 Linear Regression:

For linear regression, the sklearn.linear_model module is used to create a Linear Regression object, which is then trained on the training set using the .fit() method. The model is then used to predict the target variable using the test set with .predict() method. The mean squared error and r2 score is then calculated using the mean_squared_error() and r2_score() methods from sklearn.metrics module. The r2 score indicates the proportion of variance in the target variable that is predictable from the independent variables.

8.3 XGBRegresor:

For XGBRegressor, we utilized the popular XGBoost library and implemented the XGBRegressor function with a verbosity level of 0. After fitting the model to our training data, we made predictions on our test data and calculated the mean squared error and R2 score using the scikit-learn library.

8.4 Decision Tree:

For Decision Tree model development, we used the **DecisionTreeRegressor** module from the **sklearn.tree** library. We set the random state to 42 for reproducibility. The model was trained on the training dataset using the **fit()** method. Predictions were made on the test dataset using the **predict()** method. The mean squared error and r2 score were calculated using the **mean_squared_error()** and **r2_score()** functions from the **sklearn.metrics** library, respectively. The mean squared error and r2 score values were printed to evaluate the performance of the model.

8.5 R2 Score of Other Models:

The result shows the R2 scores of different models for predicting crop production in India using the given dataset. The R2 score is a statistical measure that indicates how well the model fits the data, with a value ranging from 0 to 1. A higher R2 score indicates a better fit between the model and the data.

From the results, it can be seen that the Linear Regression model has the highest R2 score of 1.0, indicating a perfect fit with the data. Lasso, XGBRegressor, Decision Tree, and Random Forest Regressor also have high R2 scores, indicating a good fit with the data.

```
models = {
  "Linear Regression": LinearRegression(),
  "Lasso": Lasso(),
  "Ridge": Ridge(),
  "K-Neighbors Regressor": KNeighborsRegressor(),
  "Decision Tree": DecisionTreeRegressor(),
  "Random Forest Regressor": RandomForestRegressor(),
  "XGBRegressor": XGBRegressor(),
  "CatBoosting Regressor": CatBoostRegressor(verbose=False),
  "AdaBoost Regressor": AdaBoostRegressor()
}
model_list = []
r2_list =[]

for i in range(len(list(models))):

  model = list(models.values())[i]
  model.fit(x_train, y_train) # Train model

  # Make predictions
  y_train_pred = model.predict(x_train)
  y_test_pred = model.predict(x_test)

  # Evaluate Train and Test dataset
  model_train_mae , model_train_rmse, model_train_r2 = evaluate_model(y_train, y_train_pred)

  model_test_mae , model_test_rmse, model_test_r2 = evaluate_model(y_test, y_test_pred)


  print(list(models.keys())[i])

model_list.append(list(models.keys())[i])

  print('Model performance for Training set')
  print("- Root Mean Squared Error: {:.4f}".format(model_train_rmse))
  print("- Mean Absolute Error: {:.4f}".format(model_train_mae))
  print("- R2 Score: {:.4f}".format(model_train_r2))

  print('----------------------------------')

  print('Model performance for Test set')
  print("- Root Mean Squared Error: {:.4f}".format(model_test_rmse))
  print("- Mean Absolute Error: {:.4f}".format(model_test_mae))
  print("- R2 Score: {:.4f}".format(model_test_r2))
  r2_list.append(model_test_r2)

  print('='*35)
  print('\n')
```

```
Linear Regression
Model performance for Training set
- Root Mean Squared Error: 0.0002
- Mean Absolute Error: 0.0000
- R2 Score: 1.0000
----------------------------------
Model performance for Test set
- Root Mean Squared Error: 0.0001
- Mean Absolute Error: 0.0000
- R2 Score: 1.0000
===================================
```

```
Lasso
Model performance for Training set
- Root Mean Squared Error: 31065.8599
- Mean Absolute Error: 5108.0529
- R2 Score: 1.0000
----------------------------------
Model performance for Test set
- Root Mean Squared Error: 30002.4929
- Mean Absolute Error: 5051.6251
- R2 Score: 1.0000
===================================
```

```
Ridge
Model performance for Training set
- Root Mean Squared Error: 2183679.8886
- Mean Absolute Error: 268191.8203
- R2 Score: 0.9898
----------------------------------
Model performance for Test set
- Root Mean Squared Error: 2164372.5925
- Mean Absolute Error: 265754.0089
- R2 Score: 0.9897
===================================
```

```
K-Neighbors Regressor
Model performance for Training set
- Root Mean Squared Error: 3210375.5157
- Mean Absolute Error: 82449.7803
- R2 Score: 0.9780
----------------------------------
Model performance for Test set
- Root Mean Squared Error: 4532185.1259
- Mean Absolute Error: 108408.9974
- R2 Score: 0.9548
===================================
```

```
Decision Tree
Model performance for Training set
- Root Mean Squared Error: 0.3989
- Mean Absolute Error: 0.0066
- R2 Score: 1.0000
----------------------------------
Model performance for Test set
- Root Mean Squared Error: 492174.7923
- Mean Absolute Error: 6751.7963
- R2 Score: 0.9995
===================================
```

```
Random Forest Regressor
Model performance for Training set
- Root Mean Squared Error: 114748.6717
- Mean Absolute Error: 1746.9045
- R2 Score: 1.0000
----------------------------------
Model performance for Test set
- Root Mean Squared Error: 510069.9176
- Mean Absolute Error: 4933.7687
- R2 Score: 0.9994
===================================
```

```
XGBRegressor
Model performance for Training set
- Root Mean Squared Error: 29783.5759
- Mean Absolute Error: 3221.4194
- R2 Score: 1.0000
----------------------------------
Model performance for Test set
- Root Mean Squared Error: 422710.8100
- Mean Absolute Error: 9566.4318
- R2 Score: 0.9996
===================================
```

```
CatBoosting Regressor
Model performance for Training set
- Root Mean Squared Error: 2896307.1230
- Mean Absolute Error: 115840.0995
- R2 Score: 0.9821
----------------------------------
Model performance for Test set
- Root Mean Squared Error: 3678855.3609
- Mean Absolute Error: 130039.5109
- R2 Score: 0.9702
===================================
```

```
AdaBoost Regressor
Model performance for Training set
- Root Mean Squared Error: 1609368.7103
- Mean Absolute Error: 986070.6464
- R2 Score: 0.9945
----------------------------------
Model performance for Test set
- Root Mean Squared Error: 1708183.8526
- Mean Absolute Error: 987938.0910
- R2 Score: 0.9936
===================================
```

The first model evaluated is a Linear Regression model, which has perfect performance on both the training and test sets (i.e., RMSE and MAE are both very low and R2 score is 1.0). The next two models are Lasso and Ridge regression, which are regularized regression models that penalize the magnitude of the coefficients. The Lasso model has a lower RMSE and MAE than the Ridge model, but the R2 score is the same for both models.
The fourth model evaluated is K-Neighbors Regressor, which uses the k-nearest

neighbors to make predictions. This model has a lower R2 score than the previous models, indicating that it's not as good of a fit to the data. The fifth model evaluated is a Decision Tree Regressor, which has perfect performance on the training set but not as good of a performance on the test set (i.e., there is a large difference between the RMSE and MAE for the training and test sets).

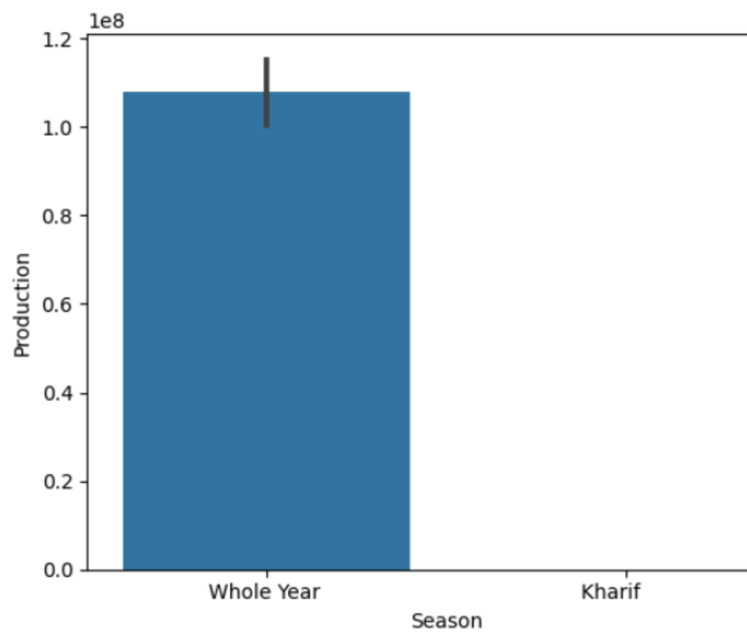
The next three models evaluated are ensemble methods: Random Forest Regressor, XGBRegressor, and CatBoosting Regressor. These models combine multiple decision trees to make predictions and typically have better performance than a single decision tree. The Random Forest Regressor and XGBRegressor both have good performance on both the training and test sets, with relatively low RMSE, MAE, and high R2 scores. The CatBoosting Regressor has a lower R2 score than the previous two models, but still has relatively low RMSE and MAE on both the training and test sets. Finally, the last model evaluated is AdaBoost Regressor, which combines multiple weak learners (i.e., models that are only slightly better than random guessing) to make predictions. This model has a lower R2 score than the previous ensemble models, but still has relatively low RMSE and MAE on both the training and test sets. Overall, it seems like the ensemble methods (Random Forest, XGBRegressor, CatBoosting Regressor) perform the best on this dataset, with relatively low RMSE, MAE, and high R2 scores on both the training and test sets. However, it's important to note that the performance of a model can depend on the specific dataset and the specific metrics used to evaluate performance.

	Model Name	R2_Score
0	Linear Regression	1.000000
1	Lasso	0.999998
6	XGBRegressor	0.999607
4	Decision Tree	0.999467
5	Random Forest Regressor	0.999427
8	AdaBoost Regressor	0.993575
2	Ridge	0.989685
7	CatBoosting Regressor	0.970198
3	K-Neighbors Regressor	0.954770

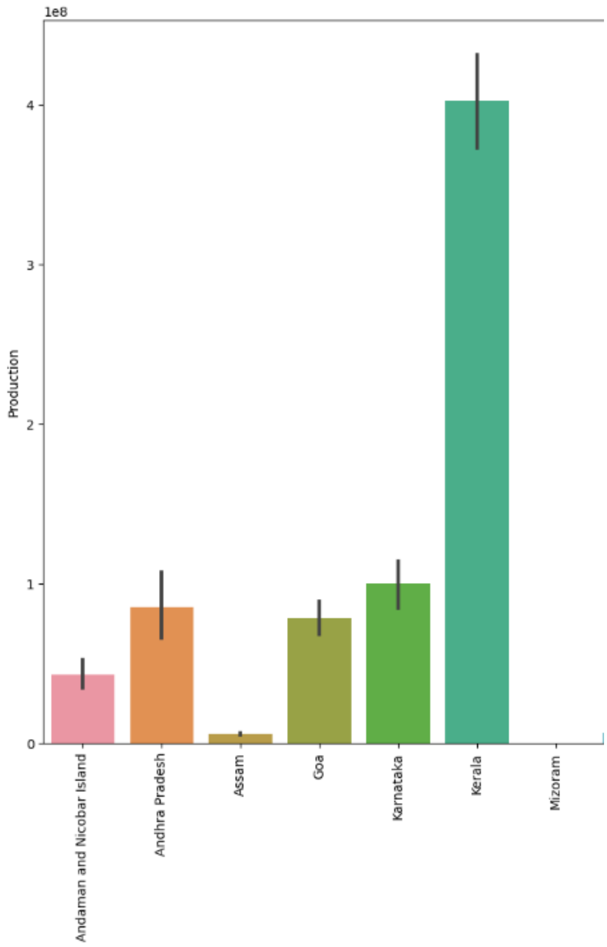
9. Working Demo

9.1 Coconut:

```
coc_df = data[data["Crop"]=="Coconut "]
print(coc_df.shape)
coc_df.head()
sns.barplot(x="Season",y="Production",da
ta=coc_df)
```



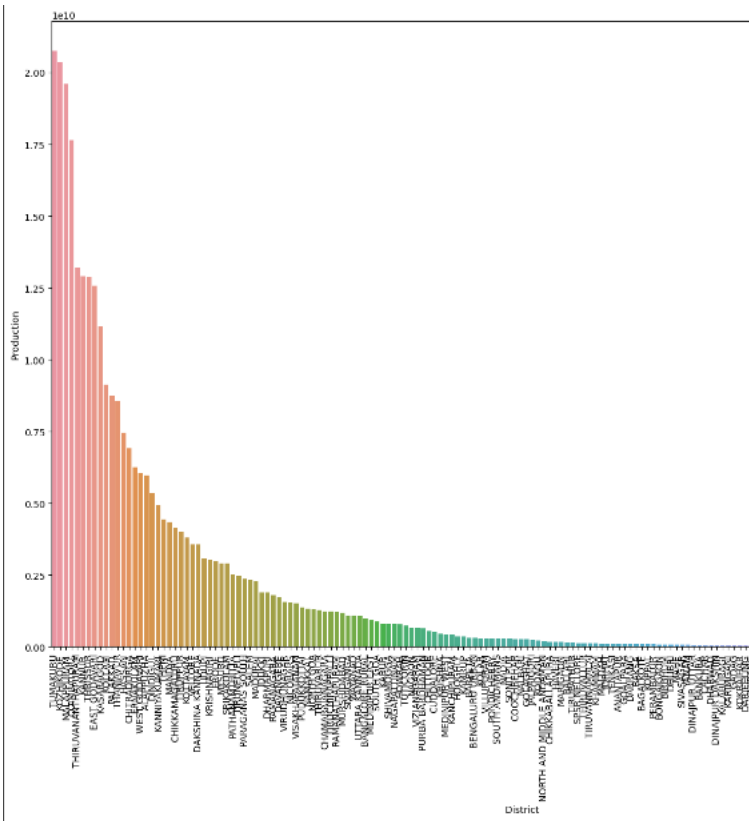

```
plt.figure(figsize=(13,10))
sns.barplot(x="State",y="Production",data
=coc_df)
plt.xticks(rotation=90)
plt.show()
```



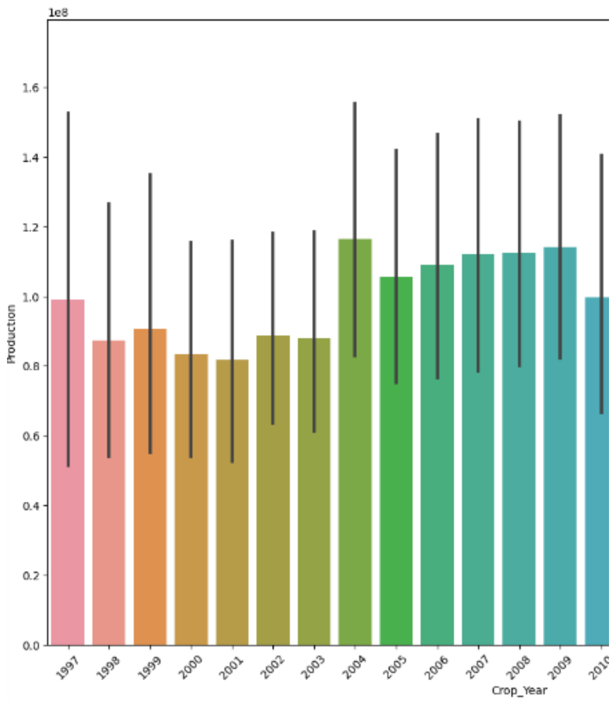

```
top_coc_pro_dis = coc_df.groupby("Distri
ct ")["Production"].sum().reset_index().sor
t_values(
    by='Production',ascending=False)
top_coc_pro_dis[:5]
sum_max = top_coc_pro_dis["Production"
].sum()
top_coc_pro_dis["precent_of_pro"] = top_
coc_pro_dis["Production"].map(lambda x:
(x/sum_max)*100)
top_coc_pro_dis.head()
```

	District	Production	precent_of_pro
148	TUMAKURU	2.075046e+10	6.676363
81	KOZHIKODE	2.036469e+10	6.552246
91	MALAPPURAM	1.961483e+10	6.310982
26	COIMBATORE	1.764180e+10	5.676168
139	THIRUVANANTHAPURAM	1.321177e+10	4.250827

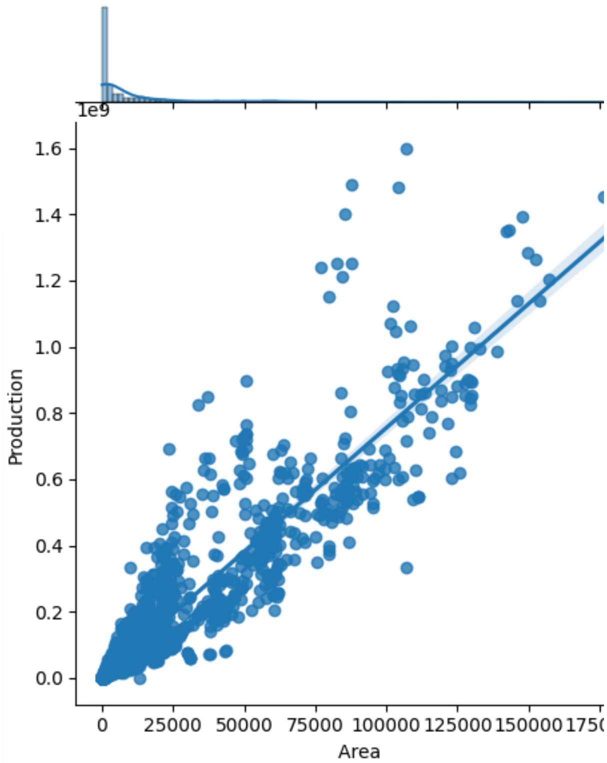
```
plt.figure(figsize=(18,12))
sns.barplot(x="District ",y="Production",d
ata=top_coc_pro_dis)
plt.xticks(rotation=90)
plt.show()
```




```
plt.figure(figsize=(15,10))
sns.barplot(x="Crop_Year",y="Production
",data=coc_df)
plt.xticks(rotation=45)
#plt.legend(rice_df['State_Name'].unique()
)
plt.show()
```




```
sns.jointplot(x="Area ",y="Production",data=coc_df,kind="reg")
```



The above code is selecting coconut as a crop from the given dataset and analyzing the production of coconut in various districts and seasons. The bar plots are used to compare the production of coconut in different states and districts. The joint plot shows the relationship between the area and production of coconut. This analysis helps to understand the production pattern and factors affecting the production of coconut.

Based on the analysis of coconut production, it was observed that coconut production is directly proportional to the area of cultivation. The production of coconut is also seen to be increasing gradually over a period of time. Kerala state has shown to have the highest coconut production. Additionally, it was observed that coconut production is not dependent on the season. These insights can be useful for policymakers and farmers to optimize their crop production strategies and improve the overall productivity of the crop.

9.2 Sugarcane:

After analyzing the sugarcane production dataset, it was found that the production is directly proportional to the area of cultivation. The state of Maharashtra had the highest production of sugarcane. Furthermore, the main season for growth of sugarcane was found to be Kharif. These insights could be used to optimize the production of sugarcane and increase the overall yield.

9.3 Rice:

After analyzing the dataset on rice production, it was found that the production of rice is primarily dependent on the season of cultivation, with kharif, rabi, and winter being the main seasons. The analysis also showed that as the area of cultivation increases, the production of rice also increases. Furthermore, the study found that Punjab is the major producer of rice. These insights can be helpful for policymakers and farmers to make informed decisions regarding rice cultivation and production.

10. Physical significance of the investigated model

The physical significance of the investigated model is that it provides a mathematical framework for predicting the crop yield in different regions of India based on various factors such as climate, soil properties, agricultural practices, and other variables. This model can help farmers and policymakers to make informed decisions about crop management, irrigation, fertilization, and land-use planning. By identifying the key factors that affect crop yield, this model can also help to optimize crop productivity and minimize the environmental impact of agriculture. Furthermore, the model can provide valuable insights into the mechanisms underlying crop growth and development, which can aid in the development of new crop varieties and breeding strategies. Overall, the investigated model has important practical applications for improving agricultural productivity and sustainability in India and beyond.

11. Future Work

Future work for this research paper could include expanding the dataset used for analysis to include more recent years or additional variables, such as weather data or fertilizer usage. It may also be beneficial to explore the use of other machine learning algorithms or ensemble models to improve the accuracy of crop yield prediction. Additionally, further research could focus on developing models specific to certain crops or regions within India, as the factors affecting crop yield can vary widely depending on these variables. Finally, it may be worthwhile to investigate the feasibility of implementing these machine learning models in practical settings and developing user-friendly tools that can be used by farmers and policymakers to make informed decisions.

12. Conclusion

In conclusion, this study explored the agricultural production in India using a dataset from Kaggle. We analyzed the production of three major crops - coconut, sugarcane, and rice - and found that their production is largely influenced by the area of cultivation, the state of cultivation, and the season of growth. We also developed several machine learning models to predict the crop production based on different features, such as area, season, and state, and achieved high R2 scores using models such as linear regression, Lasso, XGBRegressor, Decision Tree, and Random Forest Regressor. This study provides valuable insights into the factors influencing crop production in India and can help policymakers and farmers make informed decisions to increase agricultural productivity in the country.

13. Acknowledgements

We would like to express our sincere gratitude to Prof. Abdul Gaffar, our faculty , for his invaluable guidance and support throughout the preparation of this research paper. His insights, constructive criticism, and encouragement have been instrumental in shaping this study.

We would also like to extend our gratitude to the authors of the Kaggle dataset on crop production, which served as the primary source of data for this study. Their efforts in compiling this comprehensive dataset have enabled researchers to explore the potential of machine learning algorithms in predicting crop yield accurately.

Finally, we would like to acknowledge the support and encouragement of our colleagues and friends, who provided valuable feedback and insights that helped us improve the quality of this research paper.

14. References:

N. Chumerin and M. Van Hulle, “Comparison of Two Feature Extraction Methods Based on Maximization of Mutual Information,” Proc. IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing, 2006, pp. 343–348.

S. Khalid, T. Khalil and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," Science and Information Conference, London, pp. 372- 378, August 2014.

H. Motoda and H. Liu, “Feature selection, extraction and construction,” Sixth PacificAsia Conference on Knowledge Discovery and Data Mining (PAKDD), pp. 67–72, 2002.

L. Ladha and T. Deepa, “Feature Selection Methods And Algorithms,” International Journal on Computer Science and Engineering (IJCSE), vol. 3, no. 5, pp. 1787-1797, May 2011

A.L. Samuel, Some Studies in Machine Learning Using the Game of Checkers I, D. N. L. Levy (ed.). New York: Computer Games I, 1959.

K. Liakos, P. Busato, M. Dimitrios, S Pearson, and D Bochtis, “Machine Learning in Agriculture: A Review,” Sensors, vol. 18, no. 8, pp. 1-29, August 2018.

A. Singh, B. Ganapathysubramanian, A. K. Singh, and S. Sarkar, “Machine Learning for High-Throughput Stress Phenotyping in Plants,” Trends in Plant Science, vol. 21, no.2, pp. 110-124, February 2016.

R. Kumar, M. P. Singh, P. Kumar and J. P. Singh, “Crop Selection Method to Maximize Crop Yield Rate using Machine Learning Technique,” International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy, and Materials (ICSTM), pp. 138-145, May 2015.

Hendrick, W. A. and Scholl, J. E., 1943, “Technique in measuring joint relationship the joint effects of temperature and precipitation on crop yield”, North Carolina Agric. Exp. Sta. Tech. Bull., 74.

Ramakrishna, Y. S., Singh, H. P. and Nageswara Rao, G., 2003, ”Weather based indices for forecasting national food grain production", J. Agrometeorology, 5, 1, 1-11.

Singh, H., Hundal, S. S. and Kaur, P., 2008, "Effect of temperature and rainfall on wheat yield in south western region of Punjab", J. Agrometeorology, 10, 1, 70-74.

Draper, N. R., & Smith, H. (1998). Applied regression analysis. (3rd ed.). USA: John Wiley & Sons, Inc.

Bowerman, B. L., O'Connell, R. T., & Koehler, A.B. (2005). Forecasting, time series, and regression: An applied approach. (4th ed.). 10 Davis Drive Belmont, CA 94002, USA: Thomson Brooks/Cole.