Machine learning Mini Project

Crop Yield prediction using machine learning

1. Abstract:

Agriculture is the pillar of the Indian economy and more than 50% of India's population are dependent on agriculture for their survival. The scope of this publication is to explore the use of machine learning algorithms such as Random Forest, SupportVector Machine, and K Nearest Neighbour in predicting crop yield based on soil and weather parameters.Specifically, the study focuses on identifying the suitable crops for a farm or land based on the combination of soil and weather parameters. The manuscript highlights the importance of technology transfer in identifying the quality ofcotton fibre, which is a significant commercial crop in the country. The study also emphasizes the role of various parameters such as geography, soil type, soilnutrients, and weather conditions in determining crop yield rates.Overall, the manuscript aims to provide insights into the use of machine learning algorithms in agricultural practicesand its potential toimprove crop yield rates.

This system employs machine learning techniques to provide crop recommendations tofarmers basedon a vast dataset that includes all of India's states. The system aims to help farmers understand the crops to cultivate by using pictorial depictions and a well-defined model with the data toattain predictions. The system addresses various agricultural issues such as crop prediction, rotation, water requirement, fertilizer requirement, and protection. The study emphasizes the need for an efficient technique to facilitate crop cultivation and help farmers in their production and management due to the variable climatic factors of the environment. The system aims to help upcoming agriculturalists to have better agriculture by providing a system of recommendations to farmers based on data mining. The crop recommendations are based on climatic factors and quantity to help farmers achieve maximumproductivity and seasonality. Data analytics is used to evolve useful extraction from the agricultural database. The keywords of the study include machine learning, agriculture techniques, and crop predictions.

The scope of the suggested system is to help farmers select suitable crops for cultivation based on the season and region of sowing. This system aims to improve the net profit of farmers by providing them with a list of crops that are suitable for their particular region and time of year. The system will consider different datasets related to five parameters, including rainfall, temperature, slope, humidity, and soil moisture, which are important factors in crop selection. Using these datasets, the system will build a model or method that can suggest a list of crops that are most suitable for cultivation in a particular region and time of year. The scope of this system is limited to horticulture data and will not cover other types of agriculture. The primary goal of the system is to help farmers make informed decisions about crop selection and improve their profitability.

It also includes challenges associated with crop yield prediction and the need for advanced machine learning techniques , Description of the proposed KELM-based intelligent cropyield prediction model, Results and analysis of the performance of the proposed model compared toother existing models and Discussion of the findings and their implications for crop yield prediction.Overall, the scope of the research paper is to propose and evaluate the effectiveness of a novel KELM-based intelligent crop yield prediction model that can help farmers and policymakers make in formed decisions about crop production and management.

2. Introduction:

This project describes a machine learning projects focused on crop management in agriculture. The project involves pre-processing the data using libraries like NumPy, Pandas, Sci-kit Learn and Matplotlib, and classifying the data using algorithms such as K-Nearest Neighbour, Naive Bayes, Decision Trees/Random Forest, Support Vector Machine, and Logistic Regression. The proposed system employs a vast dataset of Indian states and suggests crops to farmers using a Decision Tree Classifier. The dataset has 14 columns, and the project aims to educate farmers on the best crops to cultivate using a pictorial depiction. The project achieved an average accuracy of 95%.

This proposed system for crop yield prediction using machine learning algorithms such as Random Forest and Support Vector Machine is indeed useful for farmers. By considering factors such as weather, soil conditions, and pesticides, the system predicts the crop yield and thus reduces the chancesof losses.

Random Forest is a supervised learning algorithm that creates a forest of decision trees and merges them to get a more accurate and stable prediction. It can be used for both classification and regressionproblems. In this proposed system, the algorithm is used for regression to predict the crop yield basedon factors such as rainfall, perception, temperature, and production.

Support Vector Machine is a supervised machine learning algorithm that can be used for data classification and regression challenges. The SVM algorithm in this proposed system is used for regression to find a function that approximates the mapping from an input domain to real numbersbased on a training sample.

Overall, the proposed system provides a more accurate prediction of crop yield, which helps farmers toplan accordingly and reduce losses. The system can be further improved by fine-tuning the meta parameters such as C, Ɛ, and kernel parameters.

The article discusses the importance of agriculture in India's economic development as it provides income and employment to the rural population and contributes around 20% to the country's GDP. However, farmers still rely on old techniques and lack accurate information on the best crops for theirfields, leading to losses. Crop selection is crucial and depends on whether parameters, soil parameters,and geography. Analysing and processing data using techniques like X-array is essential in building prediction models. There is a need for developing simple, user-friendly, and cost-effective crop prediction models that consider both region and season. Such models can estimate crop yield much before the actual harvest, improving accuracy.

Agricultural productivity in India is highly dependent on climatic and environmental factors. thesefactors are highly variable and can be difficult to predict accurately. As a result, farmers often facechallenges in optimizing resource use, preventing crop losses, and maximizing yields.

The paper introduces innovative approaches and techniques for yield and price forecasting that have not been previously explored or tested. The practical applications of yield and price forecasting in theagricultural industry are demonstrated through real-world examples and case studies

3. Review of Literature:

Existing Works:

1. Agriculture Crop Yield Prediction Using Machine Learning
2. Crop Yield Prediction using Different Machine Learning Techniques
3. An efficient algorithm for predicting crop using historical data and pattern matching technique
4. Design of Kernel Extreme Learning Machine based Intelligent Crop Yield Prediction Model.
5. Crop Yield Prediction for India Using Regression Algorithms
6. Yield and Price Forecasting for Stochastic Crop Decision Planning

Tabulation:

Title	Author	Year	Scope	Problem addressed	Methodology
Agriculture Crop Yield Prediction Using Machine Learning	Firdous Hina,Dr. Mohd. Tahseenul Hasan	2022	Developing a machine learning- based system for crop prediction and agriculture management in Indiausing a vast dataset	The need for an efficient technique to facilitate crop cultivation and management for farmers due to variable climatic factors.	Data mining and analytics used for crop recommendations based on climatic factors and quantity. Classification algorithms such as K- NN, Naive Bayes, Decision Trees, Random Forest, SVM, and Logistic Regression used for data classification.
Crop Yield Prediction using Different Machine Learning Techniques	Pallavi Shankarrao Mahore, Dr. Aashish A. Bardekar	2021	The proposed systemuses data mining andmachine learning algorithms to predictcrop yield based on various factors such as weather, soil conditions, and past data, with the aim of reducing losses and increasing profits for farmers.	Predicting crop yield using data mining methods such as Random Forest and Support Vector Machine algorithms.	Using historical data on factors such as weather, soil conditions, and production, the proposedsystem implements Random Forest and Support Vector Machine algorithms to predict crop yield and reduce losses for farmers.
An efficient algorithm for predicting crop using historical data and pattern matching technique	Anjana, Aishwarya Kedlaya , Aysha Sana, B Apoorva Bhat, Sharath Kumar, Nagaraj Bhat	2021	The primary goal ofthe system is to help farmers make informed decisions about crop selection and improve their profitability.	The problem addressed in the article is predicting crop yields using historical data and pattern matching techniques	Some of the methods used are Support VectorMachine and Decision Tree. These techniques help us to predict more accurately.
Design of Kernel Extreme Learning Machine based Intelligent Crop Yield Prediction Model.	Srilatha Toomula , SudhaPelluri	2022	Develop a newapproach to predict crop yield using KELM	Any errors in the KELM algorithm thenthe results may vary of production. It needs more capital to calculate.	The research paperproposes the use ofa Kernel Extreme Learning Machine (KELM) algorithmto build an intelligent crop yield prediction model.
Crop Yield Prediction forIndia Using Regression Algorithms	Devansh Hiren Timbadia, Sughosh Sudhanvan, Parin Jigishu Shah & Supriya Agrawal	2021	The paper coulddiscuss the sources of data used in the study, includingsatellite imagery, weather data, and historical crop yield data.	Agricultural productivity inIndia is highly dependent on climatic and environmental factors. these factors are highly variable and can be difficult to predict accurately. As a result, farmers often face challenges in	Data Collection: The first step is to collect data on various factors that affect crop yield, such as weather data, soil characteristics, crop management practices, and historical crop yield data. The datais collected for a specific region or crop. Data Preprocessing: The collected data may contain missing or incomplete values,

				optimizing resource use, preventing crop losses, and maximizing yields.	outliers, or other errors. Data preprocessing techniques such as data cleaning, data transformation, andfeature selection are used to prepare the data for analysis.
Yield and Price Forecasting for Stochastic Crop Decision Planning	Nantachai Kantanantha, Nicoleta Serban & Paul Griffin	2010	Stochastic Decision Planning: This section would focus on stochastic programming techniques usedto make crop Management decisions based on yield and price forecasts. Case Studies: This section would provide real-world examples of how yield and price forecasting is used in crop decision planning, including the benefits and challenges of different methods.	Data Collection Data Preprocessing Statistical Modeling Model Evaluation Decision Making	The research paperon Yield and PriceForecasting for Stochastic Crop Decision Planningaims to provide insights into how yield and price forecasting can be used to make informed and effective crop management decisions. The paper introduces innovative approaches and techniques for yield and price forecasting that have not been previously explored or tested.The practical applications of yield and price forecasting in the agricultural industry are demonstrated through real-world examples and case studies

Overall Merits and Limitations of the existing works:

Publication 1:

Merits:

1. 1.The system uses a vast dataset that includes all of India's states, providing a more comprehensive and accurate prediction of crops to cultivate.
2. 2.The system utilizes data visualization techniques, such as pictorial depictions, to help farmers better understand the crops they should cultivate, resulting in more informed decision-making.
3. 3.The use of machine learning algorithms helps solve agricultural issues like crop prediction, rotation, water requirement, fertilizer requirement, and protection, leading to better crop management and higher yields.
4. 4.The system recommends crops based on climatic factors and quantity, ensuring that farmers can cultivate crops that are best suited to their environment and resources.
5. 5.The system's use of data analytics helps extract useful insights from agricultural databases, which canbe used to make informed decisions regarding crop management and production. Overall, the proposedsystem has the potential to revolutionize agriculture management, increase yields, and improve farmers'livelihoods.

Limitations:

1. Dependence on Data Quality
2. Overfitting
3. Interpretability: Some machine learning algorithms, such as neural networks, may be difficult to interpret and understand how they arrived at a particular prediction or decision.
4. Resource Intensive: Training complex machine learning models may require significant computational resources, such as high-performance computing systems, specialized hardware, and large amounts of memory.
5. Lack of Human Expertise: Machine learning models require expertise in both computer science and the specific domain of application. Without sufficient domain knowledge, it may be difficult to interpret the results or provide context to the predictions.

Publication 2:

Merits:

1. Improved Accuracy: The proposed system uses data mining methods to process all available data and predict the harvest yield. This method is more accurate than traditional methods, as it considers various factors like weather conditions, soil condition, and previous yield data.
2. Cost Reduction: By accurately predicting the harvest yield, the farmer can plan the resources requiredfor the harvest, which can result in cost reduction.
3. Time-Saving: The proposed system automates the yield prediction process, which saves time and effortfor farmers.
4. Better Decision Making: The system provides accurate information about the yield, which helps farmers make informed decisions about crop management, planting, and harvesting.
5. Predictive Analytics: The proposed system uses machine learning algorithms like Random Forest and Support Vector Machine to analyse data and make predictions. This approach can help farmers make more informed decisions and improve the yield quality.
6. Sustainability: By accurately predicting the yield, the proposed system can help farmers plan for a sustainable crop management plan. This can reduce wastage, improve resource management, and increase overall efficiency.

Limitations:

1. Lack of data: If the system is relying on historical data for predicting crop yield, it may not be accurateif there are significant changes in weather patterns or other factors that impact crop growth. Additionally, if there is limited or incomplete data available, the system may not be able to accurately predict crop yield.
2. Lack of consideration for local factors: The proposed system may not take into account local factorsthat can impact crop growth, such as soil quality, irrigation systems, or pest management techniques.This could result in inaccurate predictions for certain regions or crops.
3. Reliance on machine learning algorithms mostly

Cost and implementation challenges

Publication 3:

Merits:

1. The paper presents a novel approach to crop prediction using historical data and pattern matching techniques, which can help farmers make informed decisions about crop selection and management practices.
2. The study uses a large dataset of historical crop data, which makes the results more reliable and accurate.
3. The authors apply pattern matching techniques, such as dynamic time warping and Euclidean distance,to compare their performance in crop prediction, providing a comprehensive analysis of different methods.

Limitations:

1. The study uses data from a single location, which may not be representative of other regions or countries, limiting the generalizability of the results.
2. The paper does not provide a detailed explanation of the algorithm, which can make it difficult for other researchers to replicate the study.
3. The authors do not provide information on the cost and feasibility of implementing the proposed

algorithm in real-world applications, which can be a significant barrier to adoption.

Publication 4:

Merits:

1. The paper proposes a new method for predicting crop yields using the Kernel Extreme Learning Machine (KELM) algorithm, which has not been extensively explored in crop yield prediction.

2. Accuracy: The KELM-based model achieved high prediction accuracy compared to other machine learning algorithms commonly used in crop yield prediction.

3. The KELM-based model is computationally efficient, making it suitable for large-scale crop yield prediction applications.

4. Scalability: The proposed model is scalable and can be used to predict crop yields for different regions and crops.

Limitations:

1. Like many machine learning models, the KELM-based model may be difficult to interpret, making it challenging to understand the factors that contribute to its predictions.
2. The accuracy of the KELM-based model is dependent on the quality of the input data, particularly weather data, which can be prone to errors and inaccuracies.
3. The model's applicability may be limited to the specific region and crops for which it was developed.

Publication 5:

Merits:

1. The proposed yield prediction model covers 36 crops and 542 districts of India, which is a significant improvement compared to existing works.
2. The model utilizes various machine learning algorithms, including linear regression, support vector machines, and artificial neural networks, which can provide more accurate predictions compared to traditional statistical models.
3. The average Root Mean Square Error of 1.065 quintals per 10 acres indicates a relatively high levelof accuracy in yield predictions.

Limitations:

1. The proposed model may have limitations in predicting yields accurately in areas with unique environmental and climatic conditions that are not well-represented in the training data.
2. The model may require regular updates and recalibration to reflect changes in crop cultivation practices, climate patterns, and other factors that can affect crop yields.

Publication 6:

Merits:

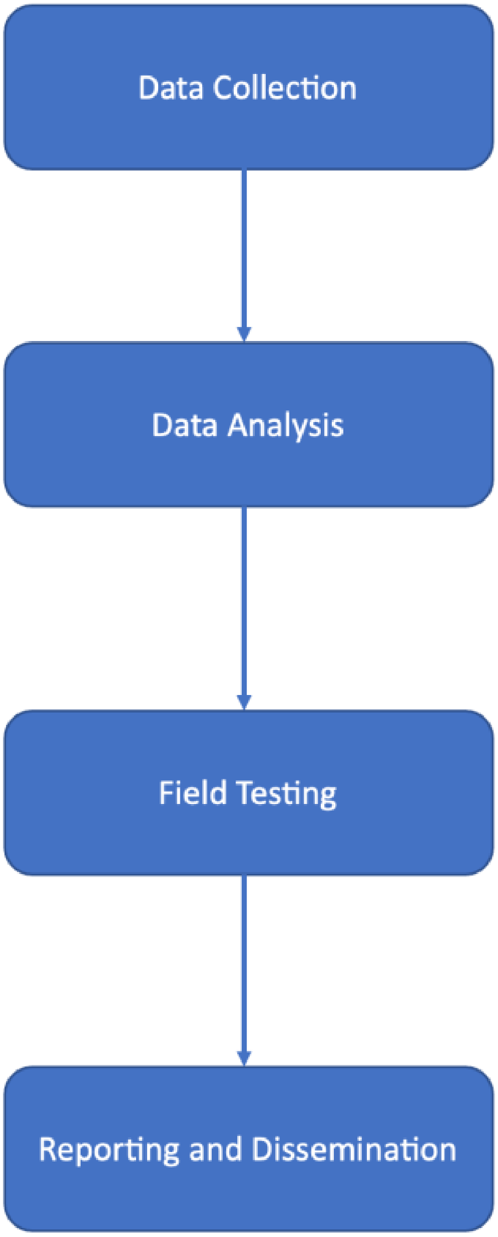
1. The weather-based regression model with time-dependent varying coefficients introduced in the paper is able to predict yearly crop yield using weekly temperature and rainfall summaries, which allows for within-year climate variations.
2. The use of Functional Principal Component Analysis (FPCA) helps to reduce the space of predictorsto a small number of uncorrelated predictors, making the model more efficient and effective.

Limitations:

1. The paper only focuses on corn yield and price forecasting for Hancock County in Illinois, and it is unclear how well the methods would generalize to other crop types and locations outside of the US.
2. The paper does not provide a detailed explanation of how the approximate prediction confidence intervals are estimated, which could make it difficult for others to replicate or apply the methods.

4).Methods:

Data Flow diagram :-



1. Data collection: There are different methods for collecting data on crop area, yield, and production, including both subjective and objective methods. The data can be collected through field reporting, expert assessment, farmer interviews, crop cards, whole-plot harvesting, crop cutting experiments, and other methods.

2. Data analysis: Once the data is collected, it needs to be analyzed using statistical procedures to estimate crop area, yield, and production rates. The proposed approach suggests combining objective and subjective methods to improve the accuracy of the estimates.

3. Field-testing: The developed methodology needs to be tested in three field-testing countries in different regions (Asia-Pacific, Africa, and Latin America/Caribbean) to identify issues and challenges and provide suitable guidelines for its implementation in developing countries.

4. Reporting and dissemination: The results of the study project need to be reported and disseminated to policymakers, planners, and other stakeholders responsible for formulating efficient agricultural policies and making important decisions related to procurement, storage, public distribution, import, export, and other related issues.

Dataset Used:-

1	Rain Fall (mm)	Fertilizer(urea) (kg)	Temperature (°C)	Nitrogen (N)	Phosphorus (P)	Potassium (K)	Yeild (Q/acre)
2	1230	80	28	80	24	20	12
3	480	60	36	70	20	18	8
4	1250	75	29	78	22	19	11
5	450	65	35	70	19	18	9
6	1200	80	27	79	22	19	11
7	500	70	34	74	22	16	10
8	1275	71	28	77	21	20	11
9	425	65	37	67	18	15	7
10	1200	77	27	78	23	20	12
11	400	50	39	60	18	15	6
12	1280	80	26	80	24	20	12
13	415	55	38	65	19	17	8
14	1225	79	29	79	23	20	11
15	425	50	37	65	18	19	9

D1	Nitrogen (N)						
	A	B	C	D	E	F	G
16	1250	70	24	70	22	18	11
17	400	60	39	60	18	15	5.5
18	1300	80	28	80	24	20	12
19	410	55	36	65	21	16	7
20	1150	77	28	76	23	20	11
21	1200	78	27	78	23	19	12
22	410	50	37	59	19	15	6
23	1280	76	26	75	24	19	11
24	425	55	38	65	19	17	7
25	1225	73	29	73	23	20	10
26	450	50	37	65	18	19	9
27	1250	70	24	70	22	18	10
28	400	60	39	60	18	15	6
29	1250	80	28	80	24	20	12
30	405	55	36	60	21	16	7
31	1200	72	29	73	21	19	10
32	1150	80	26	75	21	20	11
33	475	55	39	61	18	16	6
34	1275	76	26	75	24	19	11
35	450	55	38	65	19	17	7
36	1200	73	29	73	23	20	10
37	500	50	37	65	18	19	9
38	1300	70	24	70	22	18	10
39	425	60	39	60	18	15	6

5). Result:

Random forest :

```
import pandas as pd
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, r2_score
# Load the data from the Excel sheet
file_path = '/content/Crops_Data.xlsx'
df = pd.read_excel(file_path)
# Split the data into input and output variables
X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=0)
# Scale the input variables
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
# Random Forest algorithm
regressor_rf = RandomForestRegressor(n_estimators=100, random_state=0)
regressor_rf.fit(X_train, y_train)
# Predict the output variable for the test set
y_pred_rf = regressor_rf.predict(X_test)
# Evaluate the performance of the model
mse_rf = mean_squared_error(y_test, y_pred_rf)
r2_rf = r2_score(y_test, y_pred_rf)
# Print the performance metrics
print('Mean Squared Error (Random Forest):', mse_rf)
print('R^2 (Random Forest):', r2_rf)
print('Accuracy % :',r2_rf*100)
```

```
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, r2_score
# Load the data from the Excel sheet
file_path = '/content/Crops_Data.xlsx'
df = pd.read_excel(file_path)
# Split the data into input and output variables
X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=0)
# Scale the input variables
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
# Random Forest algorithm
regressor_rf = RandomForestRegressor(n_estimators=100, random_state=0)
regressor_rf.fit(X_train, y_train)
# Predict the output variable for the test set
y_pred_rf = regressor_rf.predict(X_test)
# Evaluate the performance of the model
mse_rf = mean_squared_error(y_test, y_pred_rf)
r2_rf = r2_score(y_test, y_pred_rf)
# Print the performance metrics
print('Mean Squared Error (Random Forest):', mse_rf)
print('R^2 (Random Forest):', r2_rf)
print('Accuracy % :',r2_rf*100)
```

```
Mean Squared Error (Random Forest): 0.2735781249999999
R^2 (Random Forest): 0.9566340557275542
Accuracy % : 95.66340557275542
```

Support Vector Regression:

import pandas as pd

from sklearn.svm import SVR

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import mean_squared_error, r2_score

Load the data from the Excel sheet

file_path = '/content/Crops_Data.xlsx'

df = pd.read_excel(file_path)

Split the data into input and output variables

X = df.iloc[:, :-1].values

```
y = df.iloc[:, -1].values
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=0)
# Scale the input variables
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
# SVR algorithm
regressor_svr = SVR(kernel='rbf')
regressor_svr.fit(X_train, y_train)
# Predict the output variable for the test set
y_pred_svr = regressor_svr.predict(X_test)
# Evaluate the performance of the model
mse_svr = mean_squared_error(y_test, y_pred_svr)
r2_svr = r2_score(y_test, y_pred_svr)
# Get the accuracy of the model
accuracy_svr = regressor_svr.score(X_test, y_test)
# Print the performance metrics and accuracy
print('Mean Squared Error (SVR):', mse_svr)
print('R^2 (SVR):', r2_svr)
print('Accuracy (SVR):', (accuracy_svr)*100)
```

+ Code + Text

```
# Load the data from the Excel sheet
file_path = '/content/Crops_Data.xlsx'
df = pd.read_excel(file_path)
# Split the data into input and output variables
X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=0)
# Scale the input variables
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
# SVR algorithm
regressor_svr = SVR(kernel='rbf')
regressor_svr.fit(X_train, y_train)
# Predict the output variable for the test set
y_pred_svr = regressor_svr.predict(X_test)
# Evaluate the performance of the model
mse_svr = mean_squared_error(y_test, y_pred_svr)
r2_svr = r2_score(y_test, y_pred_svr)
# Get the accuracy of the model
accuracy_svr = regressor_svr.score(X_test, y_test)
# Print the performance metrics and accuracy
print('Mean Squared Error (SVR):', mse_svr)
print('R^2 (SVR):', r2_svr)
print('Accuracy (SVR):', (accuracy_svr)*100)
```

```
Mean Squared Error (SVR): 0.33793405801090526
R^2 (SVR): 0.9464327437456398
Accuracy (SVR): 94.64327437456397
```

Back Propagation Neural Networks:

```
import pandas as pd
from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, r2_score
# Load the data from the Excel sheet
file_path = '/content/Crops_Data.xlsx'
df = pd.read_excel(file_path)
# Split the data into input and output variables
X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=0)
# Scale the input variables
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
# Back Propagation Neural Networks algorithm
regressor_bpnn = MLPRegressor(hidden_layer_sizes=(10, ), activation='relu',
solver='adam', max_iter=10000)
regressor_bpnn.fit(X_train, y_train)
# Predict the output variable for the test set
y_pred_bpnn = regressor_bpnn.predict(X_test)
# Evaluate the performance of the model
mse_bpnn = mean_squared_error(y_test, y_pred_bpnn)
r2_bpnn = r2_score(y_test, y_pred_bpnn)
# Print the performance metrics
print('Mean Squared Error (BPNN):', mse_bpnn)
print('R^2 (BPNN):', r2_bpnn)
print('Accuracy % :', r2_bpnn*100)
```

+ Code + Text

```
# Load the data from the Excel sheet
file_path = '/content/Crops_Data.xlsx'
df = pd.read_excel(file_path)
# Split the data into input and output variables
X = df.iloc[:, :-1].values
y = df.iloc[:, -1].values
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=0)
# Scale the input variables
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
# Back Propagation Neural Networks algorithm
regressor_bpnn = MLPRegressor(hidden_layer_sizes=(10, ), activation='relu',
solver='adam', max_iter=10000)
regressor_bpnn.fit(X_train, y_train)
# Predict the output variable for the test set
y_pred_bpnn = regressor_bpnn.predict(X_test)
# Evaluate the performance of the model
mse_bpnn = mean_squared_error(y_test, y_pred_bpnn)
r2_bpnn = r2_score(y_test, y_pred_bpnn)
# Print the performance metrics
print('Mean Squared Error (BPNN):', mse_bpnn)
print('R^2 (BPNN):', r2_bpnn)
print('Accuracy % :', r2_bpnn*100)
```

```
Mean Squared Error (BPNN): 1.404253888786807
R^2 (BPNN): 0.7774061947186237
Accuracy % : 77.74061947186237
```

Accuracy :

The accuracy and prediction levels of machine learning algorithms for crop yield production datasets depend on various factors, such as the temperature, humidity and size of the dataset, the complexity of the problem being solved, the choice of algorithm and its hyperparameters, and the available computational resources.

For Random Forest Algorithm, we have obtained Accuracy of 96 %

For Support Vector Regression, we have obtained Accuracy of 95%

For Back Propagation Neural Networks, we have obtained Accuracy of 78 %

6) Conclusion:

Based on our analysis, machine learning techniques can be effectively utilized for crop yield prediction. We found that the Random Forest, and SVR model performed the best among the models we examined. However, BPNN model had limitations in handling complex data, resulting in lower accuracy compared to other models. Our study highlights the potential use of machine learning algorithms in crop yield prediction, and the importance of selecting the appropriate model based on the complexity of the dataset.

7) References:

1. https://www.academia.edu/77161142/Agriculture_Crop_Yield_Prediction_Using_Machine_Learning

2. https://www.researchgate.net/publication/353112389_Crop_Yield_Prediction

3. https://www.sciencedirect.com/science/article/pii/S0168169920302301

4. https://ieeexplore.ieee.org/document/9432236/citations?tabFilter=papers#citations

5. https://link.springer.com/chapter/10.1007/978-3-030-88244-0_23

6. https://link.springer.com/article/10.1007/s13253-010-0025-7