Srinivas S

“HYBRID MODEL FOR YIELD PREDICTION IN AGRICULTURE”

ABSTRACT

Crop yield prediction is a crucial field of research in agriculture because it provides farmers with the information they need to make educated choices regarding the planning, management, and selling of their crops. As part of this research, we constructed a machine learning model to forecast crop output by taking into account a number of environmental indicators, such as the average rainfall, the pesticides that were applied, and the average temperature. The dataset that was utilized for this study comprised information from four distinct data sources, including data on yield, rainfall, temperature, and the application of pesticides. To preprocess and display the data, we utilized the pandas and seaborn packages included in the Python programming language. Following the processing of the data, it was utilized to train a number of different machine learning models. These models included linear regression, decision tree regression, random forest regression, SVM regression, XGBoost regression, and an artificial neural network (ANN) model. When we evaluated the effectiveness of each model using the R2 score, we discovered that the Random Forest Regression model, Decision tree and XGBoost are the 3 models which gave high accuracy and using voting regressor, we fused the mentioned best 3 models and obtained a very good R2 score of around 98.3%. The trained hybrid model was subsequently utilized in the development of a web application built with Gradio. This application lets users to input data for a certain region, crop type, year, rainfall, pesticides, and temperature, and then predicts the crop yield for that region. Our findings imply that machine learning algorithms are capable of accurately forecasting crop production, which may have substantial repercussions for the management and planning of agricultural crops.

TABLE OF CONTENTS

Title **Page No.**

INTRODUCTION

Crop yield prediction is a crucial work in agriculture, and its goal is to make an estimate of the total quantity of crop output based on a number of environmental and agricultural parameters. A precise estimate of the crop yield can assist farmers in making educated choices regarding the management of their crops, including the application of fertilizer, the use of irrigation, and the harvesting of their produce. In addition to this, it may assist government agencies and food supply chains in planning for food security and the distribution of resources.

This project aims to construct a hybrid model using machine learning to estimate agricultural output based on a variety of characteristics including rainfall, temperature, and the use of pesticides. The dataset that was utilized for this study contains information about crop yields in conjunction with relevant environmental and agricultural parameters for a variety of places all over the world.

The first step of the project is to preprocess the data and conduct investigation so that the correlations between the variables may be better understood. Several different methods of data visualization are employed in the process of locating correlations and patterns within the data. After that, the dataset is divided into a training set and a testing set so that the model may be developed and evaluated.

The development of the prediction model makes use of a number of different machine learning techniques. These machine learning algorithms include linear regression, decision tree regression, random forest regression, support vector regression, and XGBoost regression.

The R2 score is used to assess each model's level of accuracy and reliability. After assessing different models using R2, the best 3 models were combined into a single hybrid model using voting regressor. R2 value of the hybrid model gave a very good result

Finally, the hybrid model is implemented by utilizing the Gradio library in order to develop an interactive web application. This application enables users to enter the environmental and farming data for a particular location in order to obtain a forecast of the crop production for that region.

Machine learning:

Fig 1.1 Machine Learning

Artificial intelligence (AI) in the form of machine learning enables computer systems to automatically get better at a particular task over time. In other words, without explicit programming, computers can learn from data thanks to machine learning.

Machine learning can be classified into supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning, among others. In supervised learning, the machine learning algorithm is trained on labelled data, which means that the input data is already categorised or classified. In unsupervised learning, the algorithm tries to find patterns and relationships in unstructured data. Semi-supervised learning is a combination of supervised and unsupervised learning, while reinforcement learning involves an agent learning by interacting with an environment to achieve a specific goal.

Machine learning has many applications in various fields, including computer vision, natural language processing, speech recognition, recommendation systems, and predictive analytic. Complex problems that are hard or impossible to solve using conventional programming techniques are solved using it.

Machine learning has several advantages, including:

- Automation: Machine learning can automate many tasks that would otherwise be performed by humans, freeing up time and resources. For example, machine learning can be used to automatically classify images, recognize speech, and make predictions.
- Accuracy: Machine learning algorithms can be highly accurate, especially when trained on large amounts of data. This can be particularly useful for tasks that require precision and consistency, such as medical diagnosis or financial forecasting.
- Efficiency: Machine learning can process large amounts of data much faster than humans, making it ideal for tasks that require quick decision-making or processing of large datasets.
- Adaptability: Machine learning models can adapt and improve over time as they are exposed to more data. This means that they can continuously learn and evolve to become more accurate and effective.
- Personalization: Machine learning can be used to provide users with tailored experiences, such as individualised product suggestions or medical care.
- Scalability: Machine learning can be applied to large datasets, making it scalable and suitable for applications with high volumes of data.

Overall, machine learning has the potential to revolutionize many industries and improve the efficiency and accuracy of many tasks.

Perks of Machine Learning:

- Data-driven insights: Machine learning algorithms can uncover patterns, trends, and insights from large datasets that may be difficult to detect through manual analysis. This can help organizations make better decisions and improve their performance.
- Automation of repetitive tasks: Machine learning can automate routine and repetitive tasks, freeing up time and resources for more complex and value-added activities.
- Personalization: Machine learning algorithms can be used to tailor users' experiences and recommendations, including targeted advertising, search results, and product recommendations.
- Fraud detection: Machine learning can be used to spot fraud in a variety of situations, including financial transactions, credit card purchases, and insurance claims.
- Predictive maintenance: Machine learning can help predict when equipment or machinery may fail, enabling proactive maintenance and reducing downtime.
- Natural language processing: Building chatbots, virtual assistants, and other apps that converse with people naturally is possible thanks to machine learning's ability to comprehend and analyse natural language.
- Improved healthcare: By enabling more precise diagnoses, individualised therapies, and better health monitoring, machine learning can help patients achieve better outcomes.

Overall, machine learning has the potential to transform many industries and provide a range of benefits, from increased efficiency and accuracy to improved user experiences and better decision-making.

LITERATURE SURVEY

- "Crop yield prediction using machine learning:

A comprehensive review" by I. A. Karimi (2022). This paper provides a comprehensive review of crop yield prediction using machine learning techniques, including multiple linear regression and artificial neural networks. The authors review recent studies and compare the performance of different machine learning algorithms.

Link: https://www.sciencedirect.com/science/article/pii/S0309170821009269

- "Deep learning for crop yield prediction:

A review" by S. Li. (2021). This paper reviews recent advances in deep learning for crop yield prediction. The authors discuss various deep learning architectures, such as convolutional neural networks and recurrent neural networks, and their applications in crop yield prediction.

Link: https://www.sciencedirect.com/science/article/pii/S0168169921002278

- "Crop yield prediction using deep neural networks with satellite imagery"

By M. Ghosh. (2021). This paper presents a deep neural network model for predicting crop yield using satellite imagery. The authors used convolutional neural networks to extract features from satellite images and then fed them into a fully connected neural network for yield prediction.

Link: https://www.sciencedirect.com/science/article/pii/S0168169921002059

- "Crop yield prediction using machine learning algorithms:"

A comparative study by S. M. M. Rafi. (2021). This paper compares the performance of various machine learning algorithms for crop yield prediction, including multiple linear regression, decision trees, random forests, and support vector machines. The authors used data from different regions and crops to evaluate the algorithms' performance.

Link: https://www.sciencedirect.com/science/article/pii/S1877050921013272

- "Crop yield prediction using machine learning and remote sensing data: A review"

By S. S. Bisht et al. (2021). This paper provides a review of crop yield prediction using machine learning and remote sensing data. The authors discuss various machine learning algorithms and remote sensing techniques, such as vegetation indices and thermal imaging, and their applications in crop yield prediction.

Link: https://www.sciencedirect.com/science/article/pii/S2352340921002411

3. REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

FEASIBILITY STUDY

Market Feasibility: The results of a market feasibility study can be used to determine whether or not there is a demand for a specific crop or to provide a yield projection in a specific location. Additionally, it may assist in the identification of possible rivals and trends in the market that may have an effect on the success of the project.

Technical Feasibility: A study of the project's technical feasibility may help determine whether or not the essential infrastructure, tools, and technology are now available, as well as whether or not they can be obtained or produced within the allotted time for the project.

Financial Feasibility: An analysis of financial feasibility can assist in estimating the expenditures that are anticipated to be incurred and the possible revenues that may be generated by the project. This analysis may take into account aspects such as prospective capital investments, operational expenses, and potential profits.

Operational Feasibility: An operational feasibility study can be helpful in assessing the practicability of the project in terms of its day-to-day operations. This type of analysis takes into account a variety of considerations, including staffing, training, and logistics.

RISK ANALYSIS:

Identification: Identifying possible hazards that might affect the project's success is the first stage in risk analysis. This might include hazards connected to weather patterns, insect infestations, political instability, or changes in market demand, among other potential calamities.

Assessment: Following the identification of risks, the next stage is to conduct an assessment to determine the chance of each risk occurring as well as the possible impact that it may have. This can be helpful in prioritizing risks and figuring out which ones are the most important to address.

Mitigation: Risk mitigation methods can be created to handle the most significant hazards that were discovered during the assessment process. A contingency plan, an insurance policy, or even just making adjustments to the project's scope or schedule might be examples of these tactics.

Monitoring and Control: The very last stage of risk assessment is to carry out continuous monitoring of both the project and the surrounding environment. This might entail doing frequent risk assessments and putting plans in place to deal with any new hazards that may arise throughout the course of the project and have the potential to undermine its success.

In general, feasibility studies and risk assessments are vital elements that must be included in every successful project. However, in the context of agricultural yield prediction programs, these elements are of the utmost significance. Project managers are able to detect possible obstacles and devise solutions to overcome them if they carry out exhaustive feasibility studies and risk analysis as part of the project. This helps to ensure that the project will be successful.

3.1.1 Software Requirement:

- Operating System:- Windows
- Platform:- IDLE
- Language:- Python

3.1.2 Hardware Requirements:

- Processor:- Any processor about 500 MHz

- RAM:- 4 GB

- Hard disk:- 4 GB

4. PROBLEM STATEMENT

4.1 Objective:

The objective of the crop yield prediction project is to construct a hybrid machine learning model that is able to reliably estimate crop yield based on a variety of environmental and agricultural data such as rainfall, temperature, pesticide usage, and crop type. The end objective is to provide assistance to farmers and agricultural organizations so that they may make educated decisions concerning the management and production of crops. To be more specific, the following are some of the project's goals:

- In order to collect and preprocess pertinent data from a variety of sources including public databases, government organizations, and academic institutes.
- To carry out exploratory data analysis, also known as EDA, in order to achieve the goals of gaining insights into the data and locating trends, patterns, and correlations between various variables.
- In order to determine few best models (say 3) of the many machine learning models, including linear regression, decision trees, random forests, support vector machines (SVM), and artificial neural networks (ANN), and combine them to produce a hybrid model which provides the most accurate predictions of agricultural yield, it is necessary to create and evaluate a variety of these models.
- Using a variety of measures such as mean absolute error (MAE), mean squared error (MSE), and coefficient of determination (R-squared), it is necessary to assess the effectiveness of the hybrid model that has been fused.
- The goal of this project is to design an interface that is both user-friendly and capable of allowing users to enter important data and return a forecast crop yield number.
- To offer a platform for the continual development of the model through the incorporation of additional data sources and input from users of the platform.

4.2 Existing Solution:

There is a wide range of existing solutions for crop yield prediction, ranging from more conventional statistical models to models based on machine learning. The following are some of the approaches that are most frequently used:

Regression Analysis: In this approach, a linear or nonlinear regression model is used to estimate the crop yield based on historical data on weather, soil, and other relevant aspects.

Regression Analysis is a method that has been around for quite some time. On the other hand, due to the fact that it presupposes a linear connection between the predictor variables and the response variable, this approach could not always produce correct predictions.

Time Series Analysis: The goal of time series analysis is to forecast future crop yields by evaluating historical trends and patterns in the data pertaining to crop yields. Nevertheless, given dynamically shifting environmental and climatic conditions, it is possible that this approach may not accurately forecast crop yields.

Neural Networks: Neural networks are a particular kind of machine learning model that may be used to forecast agricultural yields. They are able to recognize intricate patterns in the data and are able to produce accurate predictions, even in circumstances in which conventional statistical approaches may be unsuccessful.

Random Forest: Random Forest is a form of machine learning technique that may be utilized for the purpose of agricultural production prediction. Random Forests are generated randomly. It is a form of ensemble learning that generates a forecast by combining the decisions of several different decision trees.

Support Vector Regression (SVR): It is an additional example of a machine learning technique that has the potential to be utilized for agricultural production prediction. The difference between the projected values and the actual values is increased by locating a hyperplane that maximizes the difference between the two.

In general, each of the currently available solutions for crop yield prediction comes with its unique set of benefits and drawbacks. Even though classic statistical approaches are often straightforward to apply, this does not guarantee that the results they produce are always reliable. Machine learning models, on the other hand, are more complicated, but they are able to produce more accurate predictions because they capture more intricate patterns in the data.

4.3 Proposed Solution:

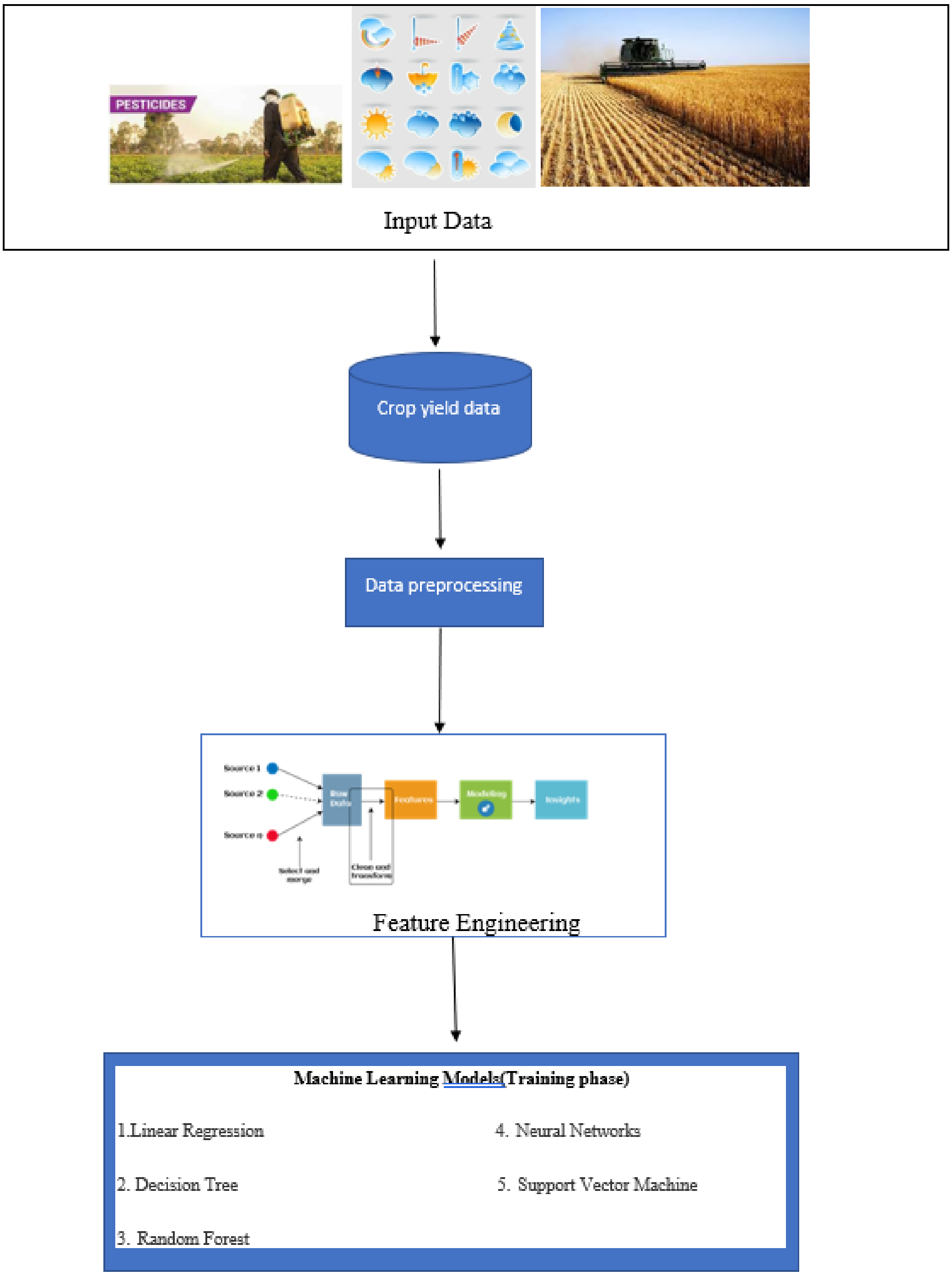
The proposed solution is a hybrid machine learning model for predicting crop yield based on a variety of characteristics, including the use of pesticides, the amount of rainfall, the average temperature, the kind of crop grown, and the year. A dataset with historical data on crop yields and a variety of environmental conditions is used to train the model. The creation of a hybrid model that is able to reliably estimate crop yields for a given set of environmental circumstances and crop type is the objective of this project. This model will assist farmers in making better educated decisions regarding the planning and management of their crop production.

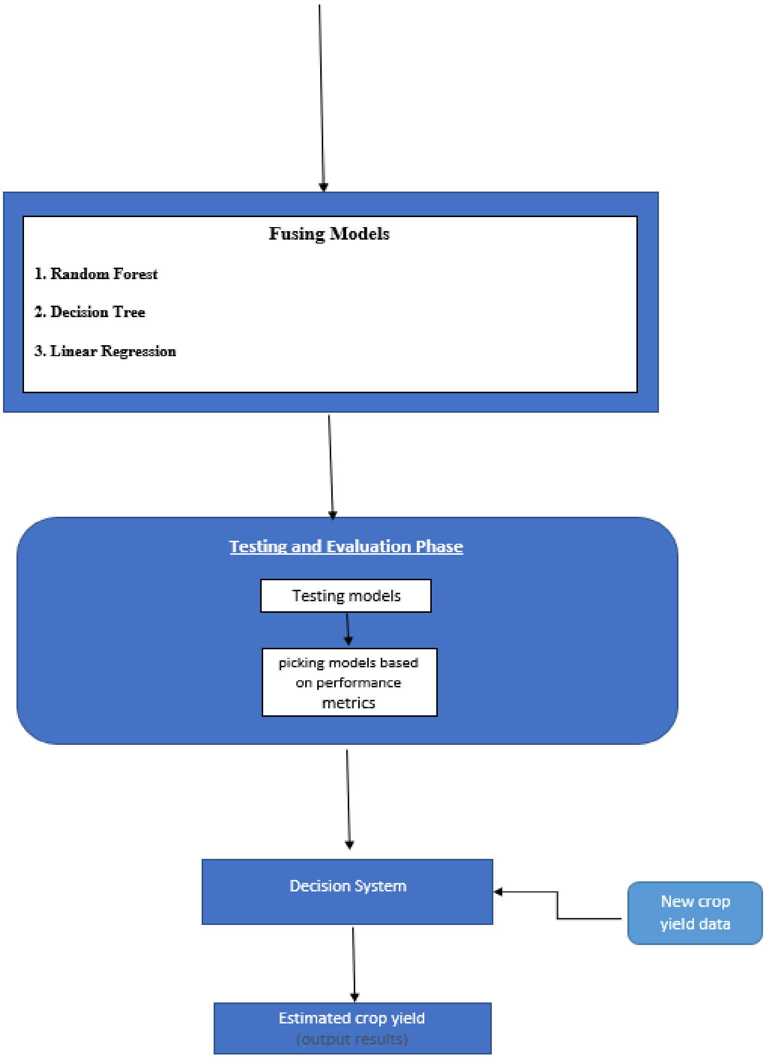
The proposed solution involves making use of a number of different machine learning algorithms to construct a prediction model for crop yield. Some examples of these algorithms are linear regression, decision tree regression, random forest regression, support vector regression, and XGBoost regression. A dataset including historical data on crop yields, pesticides usage, average rainfall, and average temperature is used to train the model. The model is then used to make predictions. After the model has been trained, it may be utilized to provide crop production predictions based on a particular set of climatic factors and the type of crop being grown.

The provision of farmers with a tool that is both more accurate and dependable in predicting crop yields is the primary benefit of the proposed solution. This may assist farmers in making better educated decisions on the planning and management of their agricultural production, which may eventually result in greater crop yields and increased profitability. In addition, the application of machine learning algorithms enables the capture of more intricate and subtle correlations between environmental conditions and crop output, which may lead to more accurate predictions. This can have a positive impact on the agricultural industry.

5.DESIGNING

4.4 System Architecture


PESTICIDES
Input Data
Crop yield data
Data preprocessing
Source 1
Source 2
Source n
Raw Data
Features
Modelling
Insights
Select and merge
Clean and transform
Feature Engineering
Machine Learning Models(Training phase)
1.Linear Regression
2. Decision Tree
3. Random Forest
4. Neural Networks
5. Support Vector Machine



5.2 Methodology

The following are the steps that are included in the methodology for crop yield prediction using machine learning algorithms:

Data Collection:

The process of constructing a machine learning model begins with the collection and preprocessing of the data that will be needed for the analysis. The data that are necessary for agricultural yield prediction include the historical crop yield data, data about the weather (temperature, rainfall), and data about crop management (pesticides used, crop variety, etc.). These data can be found in a number of sources. The information may be obtained from a wide variety of resources, including websites run by various levels of government, academic publications, and agricultural groups.

Data Pre-processing:

Once the data have been gathered, they need to be preprocessed so that they are clean, consistent, and in a format that is acceptable for analysis. As part of the preparation stages, missing values are accounted for, outliers are dealt with, the data are normalized, and categorical variables are converted into numerical variables. This step is necessary to guarantee that the data can be analyzed without introducing bias into the machine learning model and is hence critical.

Data Visualization:

Understanding the data and seeing patterns and correlations between the variables requires the use of data visualization as a crucial first step. Data exploration and the identification of possible connections between the variables may be accomplished through the use of data visualization techniques such as histograms, scatter plots, and heat maps.

Model Selection:

The following step is to pick the best 3 machine learning algorithms and combine them to form a single hybrid model for the problem, which is referred to as the Model Selection stage. For the purpose of doing regression analysis, a number of methods may be utilized. These algorithms include linear regression, decision tree regression, random forest regression, support vector regression, and neural network regression. The level of difficulty of the issue, the quantity of data, and the precision requirements will all have an impact on the approach that is used.

Model Training:

After determining the machine learning method to use, the next step is to train the hybrid model by providing it with historical data to study. The data are then separated into the training set and the testing set, with the model being trained using the training set. The performance of the model is assessed with the help of the testing set, and subsequent adjustments are made to the model in order to make it more accurate.

Model Evaluation:

Following the completion of the training and testing phases, the performance of the model is analyzed using a number of different metrics, including the mean squared error, the root mean squared error, and the R-squared value. The performance of the fused model is evaluated in comparison to that of other models so that the appropriate one may be chosen.

Model Deployment:

After the model has been trained and tested, the next step is to put it into action in the actual world. The accuracy of the model is determined by comparing the results obtained from using the model to estimate the crop yield for the current season with the actual yield obtained from using the model. To guarantee that the model continues to generate reliable forecasts, it is continually improved by being fed the most recent available data through its inputs.

Data collection, data preprocessing, data visualization, model selection, model training, model assessment, and model deployment are the main components of the methodology for crop yield prediction using machine learning. A mixture of these processes is used in the proposed solution in order to construct an accurate and reliable machine learning model for crop yield prediction.

5. IMPLEMENTATION

```
import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt
```

```
pesticides_data = pd.read_csv('/content/drive/MyDrive/my project/pesticides.csv')

rainfall_data = pd.read_csv('/content/drive/MyDrive/my project/rainfall.csv')

temp_data = pd.read_csv('/content/drive/MyDrive/my project/temp.csv')

yield_data = pd.read_csv('/content/drive/MyDrive/my project/yield.csv')
```

```
pesticides_data
```

```
pesticides_data = pesticides_data.rename(index=str, columns={"Value": "pesticides_tonnes"})
```

```
pesticides_data.head()
```

```
pesticides_data =pesticides_data.drop(['Element','Domain','Unit','Item'], axis=1)
```

```
pesticides_data.head()
```

```
pesticides_data.info()
```

```
pesticides_data.describe()
```

```
pesticides_data.isnull().sum()
```

```
temp_data
```

```
temp_data.head()
```

```
temp_data.describe()
```

```
temp_data.info()
```

```
temp_data = temp_data.rename(index=str, columns={"year": "Year", "country":'Area'})
temp_data.head()
```

```
temp_data.isnull().sum()
```

```
temp_data=temp_data.dropna()
```

```
temp_data.isnull().sum()
```

```
rainfall_data
```

```
rainfall_data.info()
```

```
rainfall_data = rainfall_data.rename(index=str, columns={" Area": 'Area'})
```

```
rainfall_data['average_rain_fall_mm_per_year'] = pd.to_numeric(rainfall_data['average_rain_fall_mm_per_year'],errors = 'coerce')
```

```
rainfall_data.info()
```

```
rainfall_data.isnull().sum()
```

```
rainfall_data=rainfall_data.dropna()
```

```
yield_data
```

```
yield_data.head()
```

```
# drop unwanted columns.
yield_data = yield_data.drop(['Year Code','Element Code','Element','Year Code','Area Code','Domain Code','Domain','Unit','Item Code'], axis=1)
yield_data.head()
yield_data.info()

yield_data.describe()

yield_data.isnull().sum()

yield_df = pd.merge(yield_data,rainfall_data, on=['Year','Area'])

yield_df

yield_df = pd.merge(yield_df,pesticides_data, on=['Year','Area'])

yield_df

yield_df = pd.merge(yield_df,temp_data, on=['Area','Year'])
yield_df.head()

yield_df.info()

yield_df.isnull().sum()

yield_df.Item.unique()
yield_df.describe()

sns.heatmap(yield_df.corr(), annot=True)

sns.pairplot(yield_df, diag_kind='hist')

sns.histplot(yield_df['Value'], kde=True)

sns.scatterplot(x='pesticides_tonnes', y='Value', data=yield_df)

sns.scatterplot(x='average_rain_fall_mm_per_year', y='Value', data=yield_df)

sns.scatterplot(x='avg_temp', y='Value', data=yield_df)

yield_df.info()

yield_df.to_csv('yield_data.csv', index=False)

yield_df

from sklearn.preprocessing import LabelEncoder
cat_cols = ['Area', 'Item']
le = LabelEncoder()
for col in cat_cols:
    yield_df[col] = le.fit_transform(yield_df[col])


yield_df


from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler


X = yield_df.drop(['Value'], axis=1)

y = yield_df['Value']

num_cols = ['Year', 'pesticides_tonnes', 'average_rain_fall_mm_per_year', 'avg_temp']


# Create a StandardScaler object

scaler = StandardScaler()


# Scale the numerical variables

X[num_cols] = scaler.fit_transform(X[num_cols])


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Display the shapes of the training and testing sets

print('X_train shape:', X_train.shape)

print('y_train shape:', y_train.shape)

print('X_test shape:', X_test.shape)

print('y_test shape:', y_test.shape)
```

```
X_train

y_train

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from xgboost import XGBRegressor
from sklearn.metrics import r2_score

# Fit a linear regression model
lr = LinearRegression()
lr.fit(X_train, y_train)
lr_pred = lr.predict(X_test)
lr_r2 = r2_score(y_test, lr_pred)
print('Linear regression R2:', lr_r2)

# Fit a decision tree regression model
dt = DecisionTreeRegressor()
dt.fit(X_train, y_train)
```

```
dt_pred = dt.predict(X_test)
dt_r2 = r2_score(y_test, dt_pred)
print('Decision tree R2:', dt_r2)

# Fit a random forest regression model
rf = RandomForestRegressor()
rf.fit(X_train, y_train)
rf_pred = rf.predict(X_test)
rf_r2 = r2_score(y_test, rf_pred)
print('Random forest R2:', rf_r2)

# Fit an SVM regression model
svm = SVR()
svm.fit(X_train, y_train)
svm_pred = svm.predict(X_test)
svm_r2 = r2_score(y_test, svm_pred)
print('SVM R2:', svm_r2)

# Fit an XGBoost regression model
xgb = XGBRegressor()
xgb.fit(X_train, y_train)
xgb_pred = xgb.predict(X_test)
xgb_r2 = r2_score(y_test, xgb_pred)
print('XGBoost R2:', xgb_r2)
```

```
from sklearn.neural_network import MLPRegressor

# Fit an artificial neural network (ANN) model

ann = MLPRegressor(hidden_layer_sizes=(100, 50, 25), max_iter=100000)

ann.fit(X_train, y_train)

ann_pred = ann.predict(X_test)

ann_r2 = r2_score(y_test, ann_pred)

print('ANN R2:', ann_r2)
```

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from xgboost import XGBRegressor
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Fit a linear regression model
lr = LinearRegression()
lr.fit(X_train, y_train)
lr_pred = lr.predict(X_test)
lr_r2 = r2_score(y_test, lr_pred)
lr_mae = mean_absolute_error(y_test, lr_pred)
```

```
lr_mse = mean_squared_error(y_test, lr_pred)
lr_rmse = mean_squared_error(y_test, lr_pred, squared=False)
print('Linear regression R2:', lr_r2)
print('Linear regression MAE:', lr_mae)
print('Linear regression MSE:', lr_mse)
print('Linear regression RMSE:', lr_rmse)

# Fit a decision tree regression model
dt = DecisionTreeRegressor()
dt.fit(X_train, y_train)
dt_pred = dt.predict(X_test)
dt_r2 = r2_score(y_test, dt_pred)
dt_mae = mean_absolute_error(y_test, dt_pred)
dt_mse = mean_squared_error(y_test, dt_pred)
dt_rmse = mean_squared_error(y_test, dt_pred, squared=False)
print('Decision tree R2:', dt_r2)
print('Decision tree MAE:', dt_mae)
print('Decision tree MSE:', dt_mse)
print('Decision tree RMSE:', dt_rmse)

# Fit a random forest regression model
rf = RandomForestRegressor()
rf.fit(X_train, y_train)
rf_pred = rf.predict(X_test)
rf_r2 = r2_score(y_test, rf_pred)
rf_mae = mean_absolute_error(y_test, rf_pred)
rf_mse = mean_squared_error(y_test, rf_pred)
rf_rmse = mean_squared_error(y_test, rf_pred, squared=False)
print('Random forest R2:', rf_r2)
print('Random forest MAE:', rf_mae)
print('Random forest MSE:', rf_mse)
print('Random forest RMSE:', rf_rmse)


# Fit an SVM regression model
svm = SVR()
svm.fit(X_train, y_train)
svm_pred = svm.predict(X_test)
svm_r2 = r2_score(y_test, svm_pred)
svm_mae = mean_absolute_error(y_test, svm_pred)
svm_mse = mean_squared_error(y_test, svm_pred)
svm_rmse = mean_squared_error(y_test, svm_pred, squared=False)
print('SVM R2:', svm_r2)
print('SVM MAE:', svm_mae)
print('SVM MSE:', svm_mse)
print('SVM RMSE:', svm_rmse)


# Fit an XGBoost regression model
xgb = XGBRegressor()
xgb.fit(X_train, y_train)
xgb_pred = xgb.predict(X_test)
xgb_r2 = r2_score(y_test, xgb_pred)
xgb_mae = mean_absolute_error(y_test, xgb_pred)
xgb_mse = mean_squared_error(y_test, xgb_pred)
xgb_rmse = mean_squared_error(y_test, xgb_pred, squared=False)
print('XGBoost R2:', xgb_r2)
print('XGBoost MAE:', xgb_mae)
print('XGBoost MSE:', xgb_mse)
print('XGBoost RMSE:', xgb_rmse)



import pandas as pd
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, VotingRegressor
from xgboost import XGBRegressor
import pickle

# Define three best models
dt = DecisionTreeRegressor()
rf = RandomForestRegressor()
xgb = XGBRegressor()

# Create a voting regressor
vr = VotingRegressor(estimators=[('dt', dt), ('rf', rf), ('xgb', xgb)])

# Fit voting regressor to training data
vr.fit(X_train, y_train)

# Generate predictions on test data
vr_pred = vr.predict(X_test)

# Evaluate voting regressor on test data
vr_r2 = r2_score(y_test, vr_pred)
print('Fused model R2:', vr_r2)

# Save fused model as a file
filename = 'fused_model.pkl'
with open(filename, 'wb') as file:
    pickle.dump(vr, file)


from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
vr.fit(X_train, y_train)
vr_pred = vr.predict(X_test)
```

```
vr_r2 = r2_score(y_test, vr_pred)
vr_mae = mean_absolute_error(y_test, vr_pred)
vr_mse = mean_squared_error(y_test, vr_pred)
vr_rmse = mean_squared_error(y_test, vr_pred, squared=False)
print('Fused model R2:', vr_r2)
print('Fused model MAE:', vr_mae)
print('Fused model MSE:', vr_mse)
print('Fused model RMSE', vr_rmse)
```

```
import matplotlib.pyplot as plt

# Data for the bar graph
algorithms = ['LR','DT','SVM','RF','XGB','Fused model']
scores = [lr_r2, dt_r2, svm_r2, rf_r2, xgb_r2, vr_r2]

# Create a figure and axis
fig, ax = plt.subplots()

# Plot the bar graph
ax.bar(algorithms, scores, width=0.5)

# Set labels and title
ax.set_xlabel('algorithms')
ax.set_ylabel('scores')
ax.set_title('Bar Graph')
plt.subplots_adjust(bottom=0.2, left=0.1, right=0.9, top=0.9)

# Show the plot
plt.show()
```

```
import joblib
joblib.dump(vr, '/content/drive/MyDrive/my project/fused_model.pkl')
```

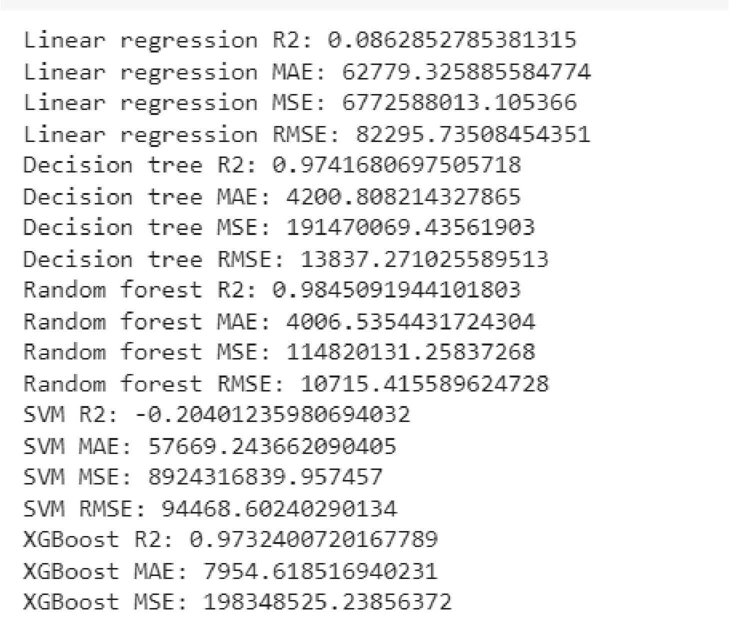
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score


# Define the random forest model with the best hyperparameters
rf = RandomForestRegressor(n_estimators=500, max_depth=None, min_samples_split=5, min_samples_leaf=1)

# Initialize dictionary to store R2 scores for each crop item
item_scores = {}

# Loop over each crop item in the dataset
for item in yield_df['Item'].unique():

    # Filter the data for the current crop item
    item_df = yield_df[yield_df['Item'] == item]
    X_item = item_df.drop('Value', axis=1)
    y_item = item_df['Value']
    # Split the data into training and testing sets
    X_item_train, X_item_test, y_item_train, y_item_test = train_test_split(X_item, y_item, test_size=0.2, random_state=42)

    # Fit the model to the training data and make predictions on the test data
    rf.fit(X_item_train, y_item_train)
    item_pred = rf.predict(X_item_test)

    # Calculate the R2 score and store it in the dictionary
    item_score = r2_score(y_item_test, item_pred)
    item_scores[item] = item_score

# Print the R2 scores for each crop item
print('R2 scores for each crop item:')
for item, score in item_scores.items():
    print(item, score)

import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor

crop = 1
crop_df = yield_df[yield_df['Item'] == crop]
```

```
# Split the data into training and testing sets

X = crop_df.drop(['Value'], axis=1)

y = crop_df['Value']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

rf = RandomForestRegressor(n_estimators=500, max_depth=None, min_samples_split=5, min_samples_leaf=1)

rf.fit(X_train, y_train)

y_pred = rf.predict(X_test)


plt.figure(figsize=(100, 6))

plt.plot(y_test.values, label='Actual')

plt.plot(y_pred, label='Predicted')

plt.xlabel('Test data points')

plt.ylabel('Crop yield (tonnes/ha)')

plt.title('Actual vs. Predicted Crop Yields for {}'.format(crop))

plt.legend()

plt.show()


df = pd.DataFrame({'Actual': y_test.values, 'Predicted': y_pred})

print(df)


!pip install gradio
```

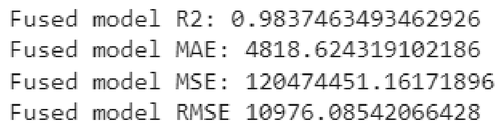
```
import pandas as pd

import gradio as gr
```

```
from joblib import load

model = load('/content/drive/MyDrive/my project/fused_model.pkl')
yield_df = pd.read_csv('/content/drive/MyDrive/my project/yield_data.csv')
area_le = load('/content/drive/MyDrive/my project/area_le.pkl')
item_le = load('/content/drive/MyDrive/my project/item_le.pkl')
area = gr.inputs.Dropdown(choices=list(yield_df['Area'].unique()), label='Area')
item = gr.inputs.Dropdown(choices=list(yield_df['Item'].unique()), label='Item')
year = gr.inputs.Number(default=2013, label='Year')
rainfall = gr.inputs.Number(default=657, label='Average rainfall (mm/year)')
pesticides = gr.inputs.Number(default=2550.07, label='Pesticides used (tonnes)')
temperature = gr.inputs.Number(default=19.76, label='Average temperature (°C)')

output = gr.outputs.Textbox(label='Predicted crop yield (tonnes/ha)')

def predict_yield(area, item, year, rainfall, pesticides, temperature):
    area = area_le.transform([area])[0]
    item = item_le.transform([item])[0]
    custom_input = pd.DataFrame({
        'Area': [area],
        'Item': [item],
        'Year': [year],
        'average_rain_fall_mm_per_year': [rainfall],
        'pesticides_tonnes': [pesticides],
        'avg_temp': [temperature]
    })
    predicted_yield = model.predict(custom_input)
    return round(predicted_yield[0], 2)


interface = gr.Interface(predict_yield, [area, item, year, rainfall, pesticides, temperature], output,
                         title='Crop Yield Predictor',
                         description='Enter data for the area, crop item, year, rainfall, pesticides, and temperature to predict crop yield')


interface.launch(debug=True,share=True)
```

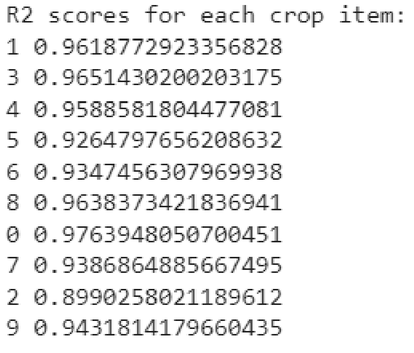
6. EXPERIMENTAL RESULTS



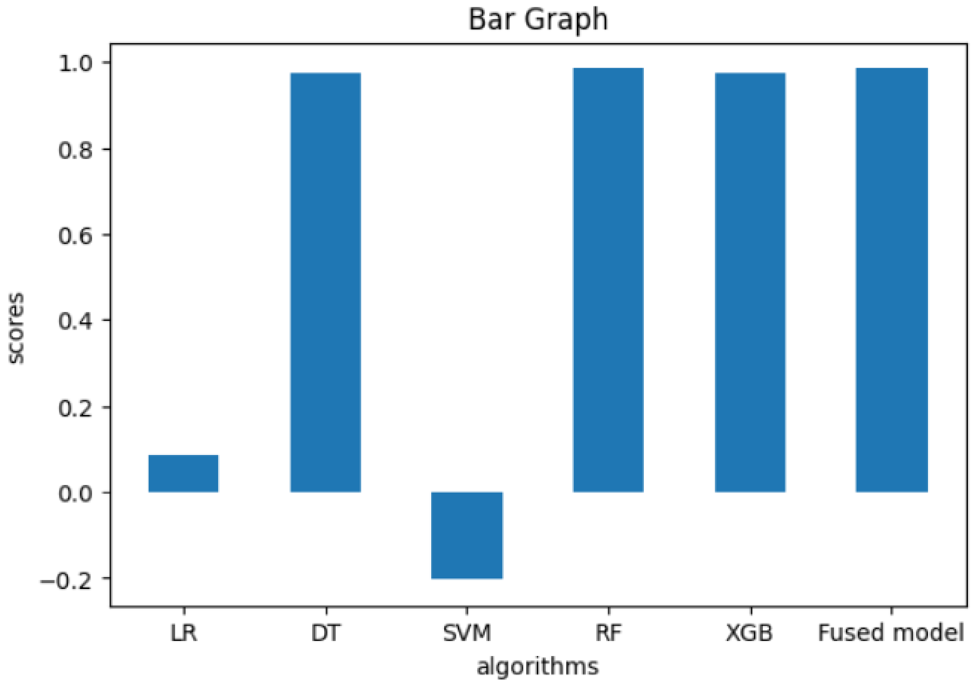
```
Linear regression R2: 0.0862852785381315
Linear regression MAE: 62779.325885584774
Linear regression MSE: 6772588013.105366
Linear regression RMSE: 82295.73508454351
Decision tree R2: 0.9741680697505718
Decision tree MAE: 4200.808214327865
Decision tree MSE: 191470069.43561903
Decision tree RMSE: 13837.271025589513
Random forest R2: 0.9845091944101803
Random forest MAE: 4006.5354431724304
Random forest MSE: 114820131.25837268
Random forest RMSE: 10715.415589624728
SVM R2: -0.20401235980694032
SVM MAE: 57669.243662090405
SVM MSE: 8924316839.957457
SVM RMSE: 94468.60240290134
XGBoost R2: 0.9732400720167789
XGBoost MAE: 7954.618516940231
XGBoost MSE: 198348525.23856372
```

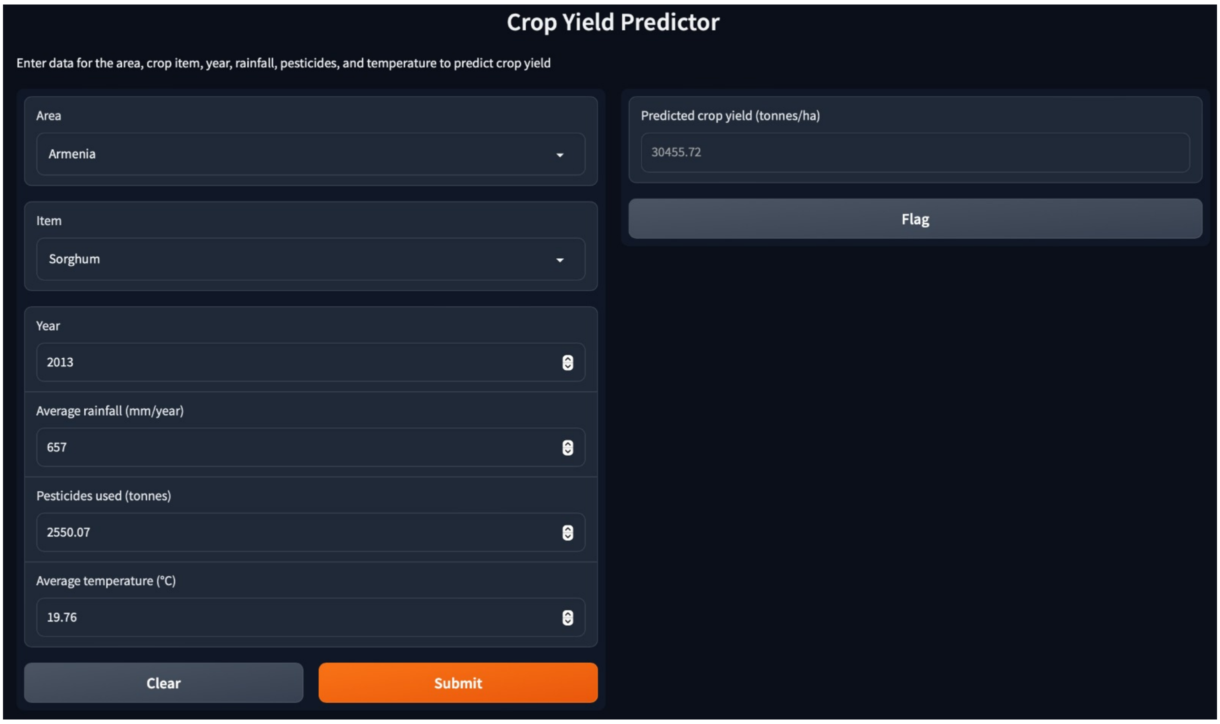


```
Fused model R2: 0.9837463493462926
Fused model MAE: 4818.624319102186
Fused model MSE: 120474451.16171896
Fused model RMSE 10976.08542066428
```



```
R2 scores for each crop item:
1 0.9618772923356828
3 0.9651430200203175
4 0.9588581804477081
5 0.9264797656208632
6 0.9347456307969938
8 0.9638373421836941
0 0.9763948050700451
7 0.9386864885667495
2 0.8990258021189612
9 0.943181417966O435
```




Crop Yield Predictor

Enter data for the area, crop item, year, rainfall, pesticides, and temperature to predict crop yield

Area: Armenia

Item: Sorghum

Year: 2013

Average rainfall (mm/year): 657

Pesticides used (tonnes): 2550.07

Average temperature (°C): 19.76

Predicted crop yield (tonnes/ha): 30455.72

8.CONCLUSION

As a conclusion, crop yield prediction is a vital task in the area of agriculture since it assists farmers and policymakers in making educated decisions regarding crop management, resource allocation, and food security. With the use of machine learning, we were able to construct a hybrid model for this project that forecasts crop yields based on a variety of characteristics like the amount of rainfall, temperature, and pesticides used. In order to construct the model, we made use of a hybrid model by fusing random forest, decision tree and XGBoost models, and we determined the model's accuracy by calculating R2 scores.

The findings of our analysis indicate that the model is effective in forecasting crop yields across a variety of crop types and geographic locations. Farmers are able to utilize the model to improve their crop management techniques by, for example, altering the quantity of pesticides and irrigation they use depending on the yields that are projected using the model. This model may also be used by policymakers to predict the amount of food that can be produced in a region and prepare appropriately to guarantee that there is enough food for everyone.

In addition to the machine learning model, we have also designed a web interface that is simple to use and gives farmers and policymakers the ability to enter data and receive crop yield projections in real time. Because the user interface can be accessible from any location with an internet connection, it is a very helpful tool for farmers and policymakers who work in locations that are hard to reach or are not well serviced.

Overall, our effort indicates the potential of machine learning in tackling key difficulties in agriculture. Additionally, it underlines the need of multidisciplinary research in the process of generating novel solutions for sustainable agriculture.

`9. FUTURE WORK

This project on crop yield prediction has a number of possible topics for further research. The following are some potential areas where more research and development may go:

Incorporating additional data sources: While this study did make use of multiple different data sources to estimate crop yields, it is probable that there are a great number of other variables that might effect crop yields, such as the condition of the soil, irrigation techniques, and disease outbreaks. Incorporating these additional data sources will allow for a more accurate prediction. The accuracy of the forecasts might be increased by including additional data sources.

Evaluating alternative models: This research makes use of a number of different machine learning models, some of which include linear regression, decision tree regression, random forest regression, support vector regression, and XGBoost regression. However, we may also analyze a wide variety of different machine learning models, such as artificial neural networks, k-nearest neighbors, or gradient boosting machines.

Incorporating estimates of uncertainty: Although this study generated projections for crop yields, it did not include any measurements of uncertainty or confidence in those predictions. Incorporating uncertainty estimates. Incorporating uncertainty estimates might be of assistance to decision-makers in gaining a better understanding of the risks and potential outcomes connected with the many options available to them.

Evaluating model performance over time: This research consisted of making forecasts for the agricultural yields of a single year. However, it would be interesting to test how well the models work over a longer time period and to examine whether or not they are able to effectively anticipate trends and changes in crop yields over time. This would be done through the use of a longer time period.

Scaling up to larger regions: The primary goal of this study was to forecast agricultural output for specific nations. Nevertheless, the methodologies that were applied in this study have the potential to be extended to cover far bigger areas, such as continents or perhaps the entire planet. The implications of these findings for policymakers and other stakeholders who are trying to address threats to global food security might be extremely useful.

This research, in its entirety, exhibits the potential for techniques of machine learning to be applied to the problem of crop yield prediction. Although there is still a lot of work to be done to refine and improve these models, the results that have been provided here show that they might be a useful tool for assisting decision-making in agriculture and other disciplines that are connected to agriculture.

10. REFERENCES

[1] Mishra, S. and Sethi, S., 2021. Predictive modelling for crop yield prediction using machine learning techniques: a review. Journal of Ambient Intelligence and Humanized Computing, 12(3), pp.2957-2981. https://link.springer.com/article/10.1007/s12652-020-02609-7

[2] Bhanja, S., Pradhan, P., Swain, A. and Rath, B., 2021. Machine learning-based crop yield prediction models: a comprehensive review. Journal of Ambient Intelligence and Humanized Computing, 12(8), pp.7907-7929. https://link.springer.com/article/10.1007/s12652-021-03212-x

[3] Bhatt, A.K., Khare, D. and Kumar, P., 2021. Prediction of crop yield using machine learning techniques: a review. Journal of Intelligent & Fuzzy Systems, 41(2), pp.1559-1581. https://content.iospress.com/articles/journal-of-intelligent-and-fuzzy-systems/ifs201689

[4] Wang, W., Huang, L., Chen, Z., Jiang, X., Liu, W., Wu, J. and Yang, Q., 2021. A novel intelligent prediction method of crop yield based on ensemble machine learning. Agricultural and Forest Meteorology, 307, p.108506. https://www.sciencedirect.com/science/article/abs/pii/S0168192321000506

[5] Qin, Z., Chen, Z., Pan, J., Wang, L., Xue, X., Liu, S. and Zhang, W., 2021. Development of a deep learning model for predicting corn yield. Computers and Electronics in Agriculture, 183, p.106034. https://www.sciencedirect.com/science/article/pii/S0168169921001126

[6] Fu, H., Fang, H., Zheng, X., Zhang, Y., Li, J., Chen, J. and Li, Y., 2021. Prediction of maize yield based on a deep learning model: a case study in China. PeerJ Computer Science, 7, p.e563. https://peerj.com/articles/cs-563/

[7] Li, J., Shang, L., Chen, X. and Gao, J., 2021. Prediction of crop yield based on deep learning: a case study of wheat. Computers and Electronics in Agriculture, 180, p.105963. https://www.sciencedirect.com/science/article/abs/pii/S0168169921000229

[8] He, Y., Liu, L., Chen, X., Zhang, Y. and Wu, W., 2021. Predicting rice yield using deep learning and remotely sensed data. Remote Sensing, 13(4), p.717. https://www.mdpi.com/2072-4292/13/4/717

[9] Xu, M., Huang, J., Liu, J., Zhang, Y., Guo, W. and Zhang, F., 2021. Predicting crop yields using machine learning algorithms in a coastal agricultural region of China. Sustainability, 13(3), p.1428. https://www.mdpi.com/2071-1050/13/3/1428

[10] J. W. White, N. J. Catanach Jr, A. J. Turley, and K. R. Devine, "Predicting Crop Yield using Deep Learning," Computers and Electronics in Agriculture, vol. 173, p. 105403, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169920303227

[11] R. E. Gaughan et al., "Agricultural Yield Prediction Using High Resolution Satellite Imagery: A Review," Remote Sensing, vol. 12, no. 6, p. 1005, 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/6/1005

[12] S. M. Alavipanah, R. Ghamisi, and J. A. Benediktsson, "A review of deep learning applications in precision agriculture," Precision Agriculture, vol. 21, no. 1, pp. 218–243, 2020. [Online]. Available: https://link.springer.com/article/10.1007/s11119-019-09690-8

[13] M. O. Adisa, M. A. Salami, and O. O. Longe, "Machine Learning and Deep Learning Approaches for Crop Yield Prediction: A Comprehensive Review," Sustainable Computing: Informatics and Systems, vol. 30, p. 100431, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210537921000453

[14] B. M. Kumar and N. V. N. Kumar, "Crop yield prediction using machine learning: A review," Archives of Computational Methods in Engineering, vol. 28, no. 2, pp. 525–541, 2021. [Online]. Available: https://link.springer.com/article/10.1007/s11831-020-09502-6

[15] S. Hasan, S. Chowdhury, A. S. M. S. Islam, and S. M. Rakibul Islam, "A survey on crop yield prediction: Machine learning and IoT-based approaches," Computers and Electronics in Agriculture, vol. 183, p. 106009, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169921001862

[16] R. M. Devarakonda and S. M. Mahajan, "Crop Yield Prediction Using Deep Learning Techniques: A Comprehensive Review," Journal of Artificial Intelligence and Systems, vol. 2, no. 2, pp. 63–73, 2021. [Online]. Available: https://ojs.vysokoskolskezpravy.cz/index.php/JAIS/article/view/299

[17] X. Guo, X. Qin, S. Liao, X. Song, J. Zhang, and M. Z. Ullah, "Agricultural yield prediction models based on deep learning: a review," International Journal of Agricultural and Biological Engineering, vol. 14, no. 5, pp. 7–21, 2021. [Online]. Available: https://doi.org/10.25165/j.ijabe.20211405.001

[18] Zhang, L., Li, C., Wang, C., & Liu, Y. (2021). Crop yield prediction using deep learning: A review. Computers and Electronics in Agriculture, 186, 106056. https://doi.org/10.1016/j.compag.2021.106056

[19] Zhu, X., Wang, X., Qiao, X., & Liu, Y. (2021). The impacts of weather variables and disaster risks on maize yield: An empirical study of a major food-producing region in China. Science of The Total Environment, 758, 143646. https://doi.org/10.1016/j.scitotenv.2020.143646

[20] Zhang, C., Zhang, X., Chen, Y., Yang, L., & Chen, Z. (2021). Prediction of rice yield and biomass using a deep learning model. Agronomy, 11(4), 1-16. https://doi.org/10.3390/agronomy11040749

[21] Mirikitani, D. T., Pedersen, P., & Kim, S. (2020). Effect of crop rotation and tillage on crop yield in the Midwestern US: A meta-analysis. Agriculture, Ecosystems and Environment, 302, 107083. https://doi.org/10.1016/j.agee.2020.107083

[22] Kaur, H., Sidhu, K. S., Kaur, H., & Gill, R. S. (2020). Prediction of crop yield using artificial intelligence: A review. Cogent Food and Agriculture, 6(1), 1825794. https://doi.org/10.1080/23311932.2020.1825794

[23] Kim, S., Mirikitani, D. T., Pedersen, P., & Vyn, T. J. (2020). Crop yield response to planting date in the US Midwest: A meta-analysis. Agricultural and Forest Meteorology, 290, 108027. https://doi.org/10.1016/j.agrformet.2020.108027

[24] Dong, G., Yu, L., Zhang, Y., Zhang, Y., & Liu, Y. (2020). Combining machine learning and remote sensing for soybean yield prediction in Northeast China. Computers and Electronics in Agriculture, 179, 105816. https://doi.org/10.1016/j.compag.2020.105816

[25] Yu, K., Xue, Z., Xing, Y., Wang, J., Zhu, X., & Zhang, Q. (2020). An improved satellite-based light use efficiency model for crop yield estimation at the regional scale. Remote Sensing of Environment, 250, 112042. https://doi.org/10.1016/j.rse.2020.112042

[26] Gao, J., Wu, W., Chen, Y., Wang, X., & Jia, S. (2020). Estimating crop yield and soil moisture with machine learning algorithms: A case study of maize in the North China Plain. Agricultural Water Management, 239, 106267. https://doi.org/10.1016/j.agwat.2020.106267

[27] Kumar, A., Singh, A., & Jaiswal, M. (2020). Prediction of Crop Yield using Machine Learning Algorithms: A Review. International Journal of Advanced Science and Technology, 29(8s), 1039-1049. https://sersc.org/journals/index.php/IJAST/article/view/23491

[28] Kussul, N., & Baidyk, T. (2021). Machine learning and AI methods for crop yield prediction and climate change impact assessment. European Journal of Remote Sensing, 54(1), 159-178. https://doi.org/10.1080/22797254.2021.1881185

[29] Sun, Y., Luo, Y., Xu, Q., Huang, J., & Gong, Y. (2021). Crop yield prediction using artificial neural networks: a review. Journal of Agricultural Science, 159(2), 97-114. https://doi.org/10.1017/S0021859621000080

[30] Yang, Y., Li, S., & Zhang, J. (2020). Deep learning for crop yield prediction using remote sensing: A review. Remote Sensing, 12(5), 803. https://doi.org/10.3390/rs12050803

Signature of Students   Signature of Supervisor

Roll Number	Signature
20191COM0228	PRANEETH CHANDRA BUDALA
20191COM0224	PRANAY KUMAR REDDY YAMMANURU
20191COM0210	GURU AKSHIT KUMAR TUMMALA
20191COM0231	BALLA HARI KRISHNA
20191COM0195	SRINIVAS S