

**MEKDELA AMBA UNIVERSITY**

**College of computing and Informatics**

**Department of Software Engineering**

**AI Group Project Report**

**Crop Yield Prediction**

**Group Four**

|  |  |  |
| --- | --- | --- |
| **No** | **Name** | **ID** |
| **1** | TAREKEGN ABEBE FERENJO | MAU/1402050 |
| **2** | TEMESGEN MINWYELET | MAU/1402310 |
| **3** | TIBEBE DIFABACHEW BELETE | MAU/1402341 |
| **4** | YENEW DAREGOT MENIKIR | MAU/1402494 |
| **5** | YOHAS HABTAMU BARASA | MAU/1402527 |
| **6** | YONAS SAHILE WELDETSADIK | MAU/1402530 |
| **7** | TSION TAFESE DEBOCH | MAU/1402392 |
| **8** | TSION WENDEMU DEJENE | MAU/1402398 |
| **9** | REKIK LEGESE TIRFE | MAU/1402050 |

**Submitted to:**   **Mr.Chalachew (MSc)**

**May, 2017**

**Mekdela Amba, Ethiopia**

# Summary

This report details the development of a machine learning model to predict crop yield based on various climatic factors. The project leverages environmental factors such as precipitation, humidity, and temperature to predict crop yields for different agricultural products, including cocoa beans, oil palm fruit, rice paddy, and natural rubber. Multiple regression models were evaluated, with Random Forest and XGBoost demonstrating strong predictive performance. A Streamlit web application was developed to provide an interactive interface for users to input parameters and receive yield predictions. The process includes data loading and preprocessing, exploratory data visualization, model selection, training, evaluation, and the development of a Streamlit application for user interaction. The Random Forest Regressor and Decision Tree Regressor models were evaluated, with the Random Forest Regressor showing superior performance. The Streamlit app allows users to input climatic parameters to predict crop yield.

# INTRODUCTION

Accurate Crop yield prediction is crucial for agricultural planning, food security, and economic forecasting and resource management. Traditional methods rely on historical trends and expert knowledge, but machine learning offers a data-driven approach by analyzing environmental factors. This project aims to develop a predictive model using machine learning techniques to estimate crop yield based on climatic data such as precipitation, humidity, and temperature. It explores the use of regression models to predict crop yields based on climate data. The model's development includes several stages: data preprocessing, model training, and evaluation.. The system was deployed as a user-friendly web app using Streamlit.

# Dataset and Preprocessing

The dataset used in this analysis contains records of crop yield and corresponding climatic conditions.

### ****Dataset Overview****

**Features**:

* **Crop**: Type of crop (e.g., cocoa beans, oil palm fruit).
* **Precipitation (mm/day)**: Daily rainfall.
* **Specific Humidity (g/kg)**: Moisture content in the air.
* **Relative Humidity (%)**: Percentage of moisture relative to saturation.
* **Temperature (°C)**: Air temperature at 2 meters.
* **Yield**: Target variable (crop output).

**Preprocessing steps**

1. **Label Encoding:** The categorical ‘Crop’ column was encoded using Label Encoding to convert it into a numerical format suitable for machine learning models.

1. **Data Splitting:** The dataset was divided into training and testing sets to properly train and evaluate the models.
2. **Feature Scaling:** Standardization was applied to the numerical features to ensure that all features contribute equally to the model training process.

# Data Visualization

Although the provided code includes data visualization (such as bar plots and scatter plots), it primarily focuses on model performance rather than exploratory data analysis. For instance, a bar plot is used to compare the R² scores of different models. Additionally, scatter plots are used to visualize actual vs. predicted yield.

**Yield Distribution**: Histograms and box plots were used to analyze yield variations across crops.

**Correlation Heatmap**: Explored relationships between climate variables and yield.

**Actual vs. Predicted Plots**: Scatter plots with a reference line (y = x) visualized model performance.

# Machine Learning Model

### ****Model Selection****

Five regression models were evaluated:

**Random Forest** (Best R²: ~0.92)

**XGBoost** (R²: ~0.90)

**Lasso Regression** (R²: ~0.75)

**Decision Tree** (R²: ~0.85)

1. **Nearest Neighbors (KNN)** (R²: ~0.70)

The Random Forest Regressor and Decision Tree Regressor were chosen for further analysis due to their strong performance.

**Model Training and Evaluation:**

The models were trained on the training dataset, and their performance was evaluated using Mean Squared Error (MSE) and R² Score on the test dataset. The Random Forest Regressor consistently outperformed other models, demonstrating higher accuracy and less overfitting.

**Metrics**:

* **Mean Absolute Error (MAE)**: Measures average prediction error.
* **Mean Squared Error (MSE)**: Penalizes larger errors.
* **R² Score**: Indicates variance explained by the model.

**Results**:

* Random Forest and XGBoost outperformed linear models due to their ability to capture non-linear relationships.
* Decision Trees showed signs of overfitting, mitigated by hyper parameter tuning.

# Model Optimization Techniques

The primary optimization technique used was hyperparameter tuning. For example, the number of estimators in the Random Forest Regressor and the maximum depth of the Decision Tree Regressor were adjusted to improve performance and reduce overfitting.

* **Hyper parameter Tuning**:

**Random Forest**: Adjusted n\_estimators and max\_depth.

**Decision Tree**: Optimized min\_samples\_split and min\_samples\_leaf.

* **Cross-Validation**: Ensured robustness by evaluating performance across multiple splits.
* **Feature Importance**: Random Forest identified precipitation and temperature as key predictors.

# Streamlit App Development

**App Architecture:**

**Frontend**: Streamlit for interactive UI.

**Backend**: Pre-trained Random Forest and Decision Tree models.

The Streamlit application is designed to provide a user-friendly interface for predicting crop yield. The architecture includes:

1. **Data Loading and Preprocessing:** Loading the dataset and performing necessary preprocessing steps.
2. **Model Training:** Training the selected machine learning models.
3. **Prediction Interface:** Allowing users to input climatic parameters.
4. **Result Display:** Displaying the predicted crop yield.

**Features and UI Interface:**

* **Input Fields:** The app includes input fields for users to enter values for crop type, precipitation, specific humidity, relative humidity, and temperature.
* **Prediction Display:** The predicted yield is displayed clearly upon submission of the input parameters.
* **Model Performance Metrics:** The app also displays the MSE and R² Score for the trained models.
* **Visualization:** Users can visualize the Decision Tree structure and the comparison between actual and predicted yield.

**Model Metrics Display**: Shows MSE and R² scores.

**Prediction Interface**: Users input climate parameters to get yield forecasts.

**Visualizations**:

Actual vs. Predicted plots.

Decision Tree structure (optional).

### ****UI Interface****

**Input Fields**: Numeric inputs for climate variables.

**Output**: Predicted yield from both Decision Tree and Random Forest.

**Interactive Plots**: Matplotlib visualizations embedded in the app.

# Challenges and Solutions

## Challenges

* Ensuring the accuracy and consistency of the dataset
* Overfitting
* Creating a user-friendly interface
* High variance in Decision Tree predictions
* Categorical crop names
* Feature scaling requirements

## Solutions

* Challenge of Ensuring the accuracy and consistency of the dataset was addressed through careful data cleaning and preprocessing.
* Challenge of Overfitting was mitigated by tuning hyperparameters and using ensemble methods like Random Forest Regressor.
* Challenge of Creating a user-friendly interface was achieved through iterative design and testing.
* Used Random Forest for better generalization for High variance in Decision Tree predictions
* Applying Label Encoding for Categorical crop names
* Standardized data for sensitive models for the challenge of Feature scaling requirements.

# Conclusion

The crop yield prediction model, particularly the Random Forest Regressor, provides accurate and reliable predictions based on climatic factors. The Streamlit application offers a practical tool for farmers and agricultural experts to estimate crop yield, aiding in better decision-making and resource allocation.

* Random Forest and XGBoost were the most effective models for crop yield prediction.
* The Streamlit app provides an accessible tool for farmers and agronomists.

# References

**1. Lobell, D. B., & Burke, M. B. (2010).**On the use of statistical models to predict crop yield responses to climate change.  
Agricultural and Forest Meteorology, 150(11), 1443-1452.  
DOI: [10.1016/j.agrformet.2010.07.008](https://doi.org/10.1016/j.agrformet.2010.07.008)

**2.Van Klompenburg, T., Kassahun, A., & Catal, C. (2020).**  
Crop yield prediction using machine learning: A systematic literature review.  
Computers and Electronics in Agriculture, 177, 105709.  
DOI: [10.1016/j.compag.2020.105709](https://doi.org/10.1016/j.compag.2020.105709)

**3.Scikit-learn Documentation**  
Random Forest, Decision Trees, and other regression models.  
<https://scikit-learn.org/stable/>

**4.Streamlit Documentation**  
Building interactive web apps for machine learning.  
<https://docs.streamlit.io/>

**5.Kaggle Datasets**  
<https://www.kaggle.com/datasets>