Industrial Internship Report on

Prediction of Agriculture Crop Production in India

Prepared by

Anushree

Executive Summary

This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT).

This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks' time.

My project was Prediction of Agriculture Crop Production in India.

This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship.

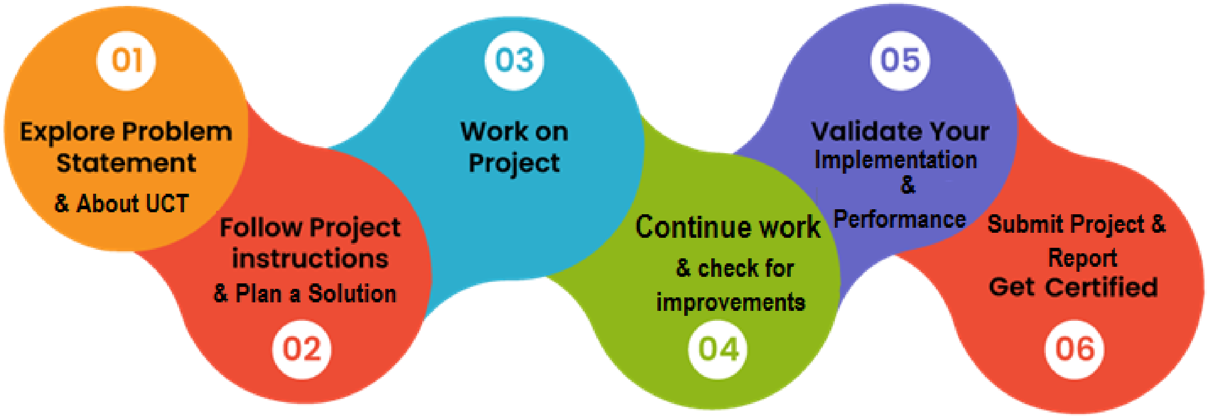
TABLE OF CONTENTS

1 Preface

The experience of working with the USC/UCT team fully was really great, the team and the mentors were actually into this journey of six weeks internship Opportunity given by USC/UCT.

I had been explored to evolving technology Data Science and Machine Learning during this six-week internship opportunity given by USC/UCT.

The program was all well planned. The USC/UCT gave the real- world experience about the Machine learning and guided throughout the internship.



I had gone through all the e-resources provided by the organization and utilized every video sessions to upskill myself and serve the internship provided by USC/UCT.

Thanks to Nitin Tyagi Apurv sir who have helped me throughout this journey.

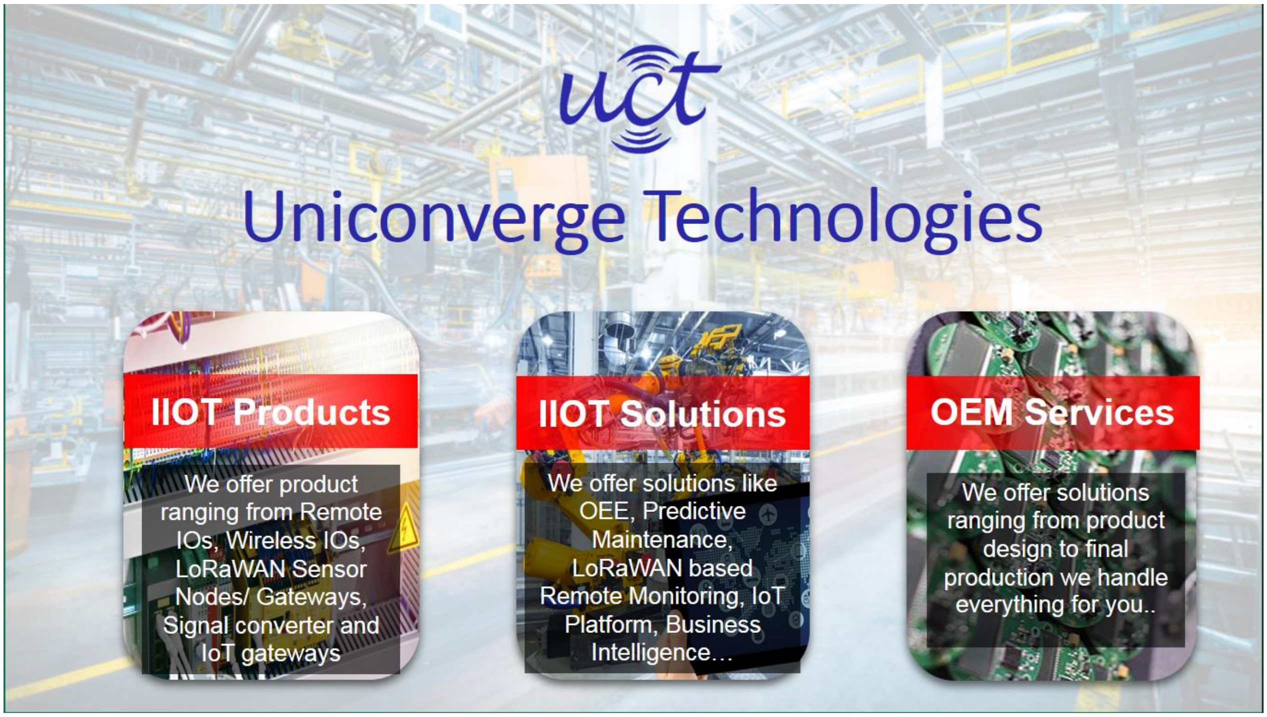
I would highly recommend this internship to all my juniors and peers to upskill themselves through real world experience provided by the USC/UCT.

2 Introduction

2.1 About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various **Cutting Edge Technologies e.g. Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end** etc.



uct

Uniconverge Technologies

IIOT Products

We offer product ranging from Remote IOs, Wireless IOs, LoRaWAN Sensor Nodes/ Gateways, Signal converter and IoT gateways

IIOT Solutions

We offer solutions like OEE, Predictive Maintenance, LoRaWAN based Remote Monitoring, IoT Platform, Business Intelligence…

OEM Services

We offer solutions ranging from product design to final production we handle everything for you..

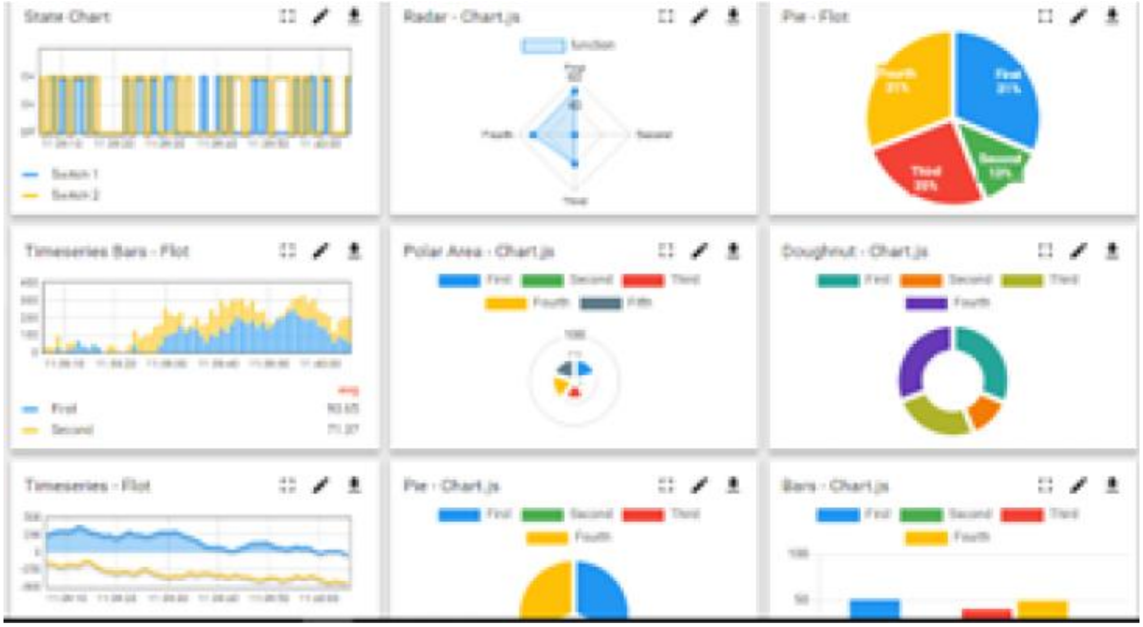
i. UCT IoT Platform (uct Insight)

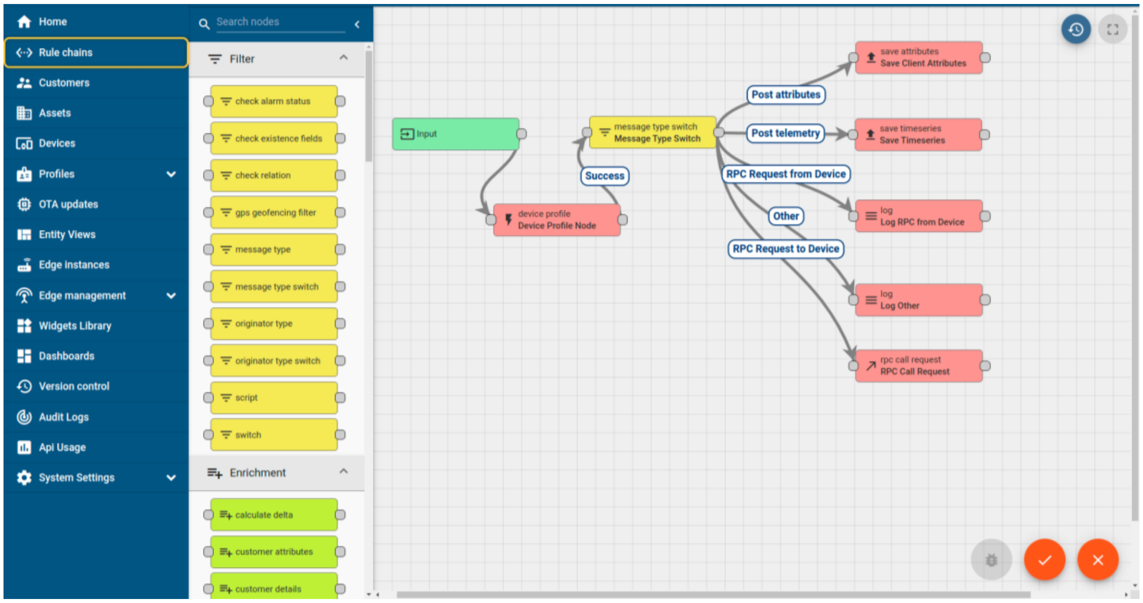
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable “insight” for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
- It supports both cloud and on-premises deployments.

It has features to

- Build Your own dashboard
- Analytics and Reporting
- Alert and Notification
- Integration with third party application(Power BI, SAP, ERP)
- Rule Engine





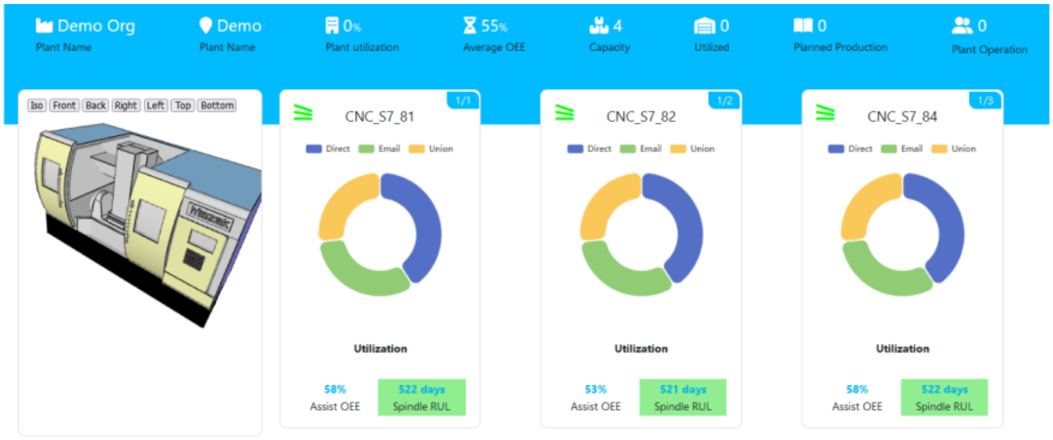
ii. Smart Factory Platform (FACTORY WATCH)

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

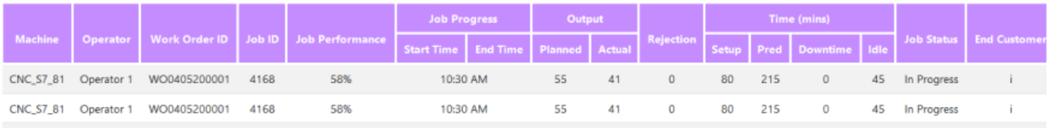
- with a scalable solution for their Production and asset monitoring
- OEE and predictive maintenance solution scaling up to digital twin for your assets.
- to unleased the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.

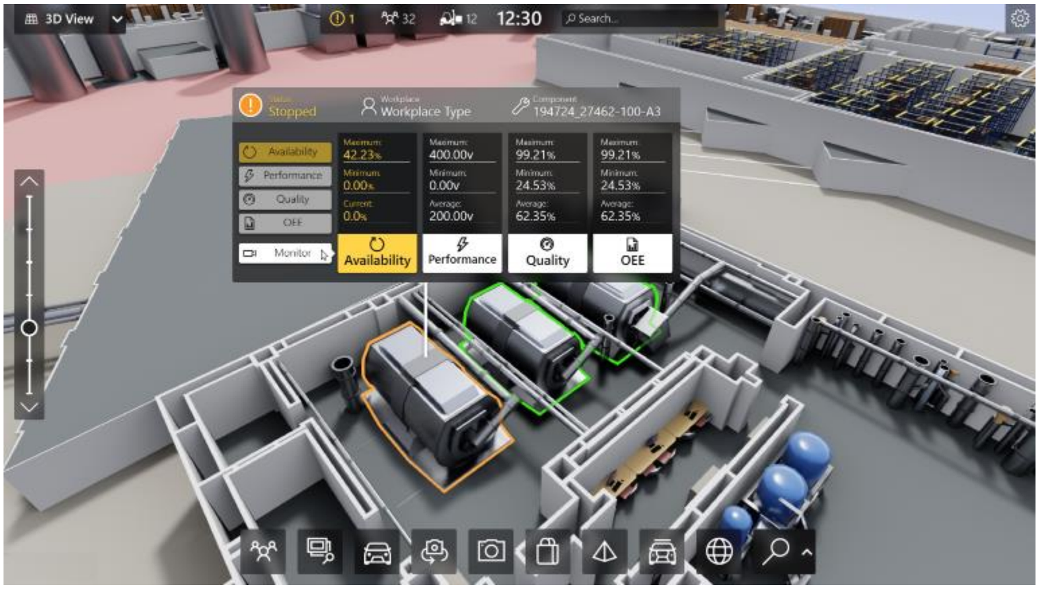


Demo Org Plant Name | Demo Plant Name | 0% Plant utilization | 55% Average OEE | 4 Capacity | 0 Utilized | 0 Planned Production | 0 Plant Operation

Machine	Utilization	Assist OEE	Spindle RUL
CNC_S7_81		58%	522 days
CNC_S7_82		53%	521 days
CNC_S7_84		58%	522 days



Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i



Status: Stopped
Workplace: Workplace Type
Component: 194724_27462-100-A3

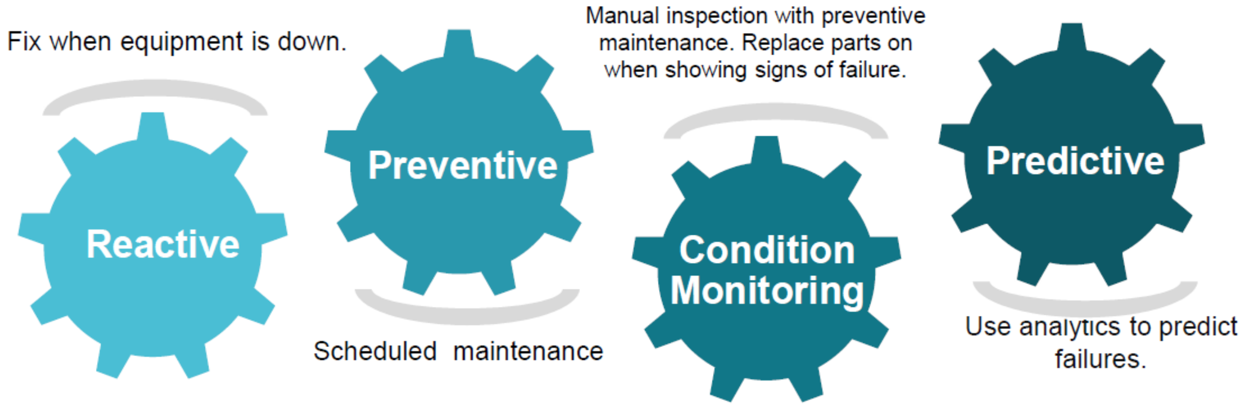
Availability	Performance	Quality	OEE
Maximum: 42.23%	Maximum: 400.00v	Maximum: 99.21%	Maximum: 99.21%
Minimum: 0.00%	Minimum: 0.00v	Minimum: 24.53%	Minimum: 24.53%
Current: 0.0%	Average: 200.00v	Average: 62.35%	Average: 62.35%

iii. LoRaWAN based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv. Predictive Maintenance

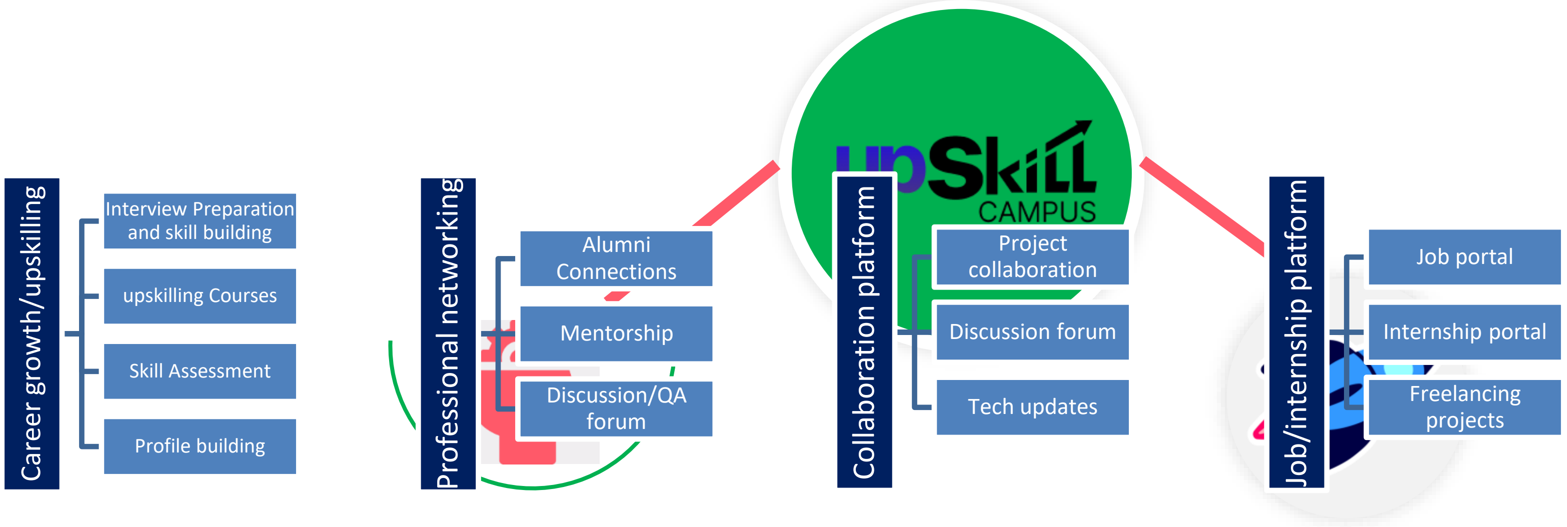
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.


Fix when equipment is down.
Reactive
Preventive
Scheduled maintenance
Manual inspection with preventive maintenance. Replace parts on when showing signs of failure.
Condition Monitoring
Predictive
Use analytics to predict failures.

2.2 About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



https://www.upskillcampus.com/

Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career

upSkill Campus aiming to upskill 1 million learners in next 5 year

2.3 The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

Objectives of this Internship program

The objective for this internship program was to

- ☛ get practical experience of working in the industry.
- ☛ to solve real world problems.
- ☛ to have improved job prospects.
- ☛ to have Improved understanding of our field and its applications.

☛ to have Personal growth like better communication and problem solving.

2.4 Reference

[1] The reference for the data set was taken by the Kaggle website.

[2] The lecture and video sessions in the project plan provided by UCT was upto the mark to learn about the data science and machine learning

[3] The e-resources provided in the week 2 till week 5 were good enough to learn about the Python and data science concepts.

2.5 Glossary

Terms	Acronym
Dataset	DST
Preprocessing	PP
Training Model	TM
Testing Model	TESM
Random forest regressor	**RFG**

3 Problem Statement

Agriculture is a critical sector in India, contributing significantly to the country's economy and food security. However, predicting crop production is a complex task influenced by various factors such as weather conditions, soil quality, irrigation practices, pest infestations, and the use of fertilizers. Accurate predictions of crop production can help farmers, policymakers, and stakeholders make informed decisions, optimize resource allocation, and enhance food security.

The objective of this project is to develop a machine learning model that predicts the production of various crops in different regions of India. The model will utilize historical data on crop yields, weather patterns, soil conditions, and other relevant agricultural data. By analyzing this data, the model aims to provide accurate and timely predictions that can assist in strategic planning and decision-making in the agricultural sector.

4 Existing and Proposed solution

Traditional Statistical Models:

Linear Regression: Often used for predicting crop yields based on historical data, weather conditions, and other factors. However, linear regression may not capture the complex relationships between variables in agricultural data.

Time Series Analysis: Techniques such as ARIMA (Auto Regressive Integrated Moving Average) models predict future crop yields based on past trends. These methods can struggle with handling non-linearities and multiple influencing factors.

Enhanced Prediction System Using Random Forest Regressor and Decision Tree

- Develop an accurate and robust machine learning model using Random Forest Regressor and Decision Tree to predict agricultural crop production in India.
- Integrate diverse data sources, including historical crop yields, weather data, soil information, and satellite imagery.
- Provide actionable insights and predictions to farmers, policymakers, and stakeholders.

4.1 Code submission (Github link):

https://github.com/Anu123shree/upskillcampus/blob/main/crop_production_prediction.pdf

4.2 Report submission (Github link):

https://github.com/Anu123shree/upskillcampus/blob/main/Prediction_of_agriCropProduction_prediction_Anushree_USC_UCT.docx

5 Proposed Design/ Model

Key Components of the Proposed Solution:

Data Collection:

- Historical Crop Data: Collect historical data on crop yields for various regions in India from government databases and agricultural research institutions.
- Weather Data: Integrate weather data (temperature, rainfall, humidity, etc.) from meteorological departments and weather stations.
- Soil Data: Gather soil quality data (pH, nutrient levels, moisture content) from agricultural surveys and soil testing labs.
- Remote Sensing Data: Utilize satellite imagery and indices such as NDVI for real-time monitoring of crop health.

Data Preprocessing:

- Clean and preprocess the collected data to handle missing values, outliers, and inconsistencies.
- Perform feature engineering to create meaningful input variables (e.g., lagged weather variables, soil quality indices).

Model Development:

- Exploratory Data Analysis (EDA): Conduct EDA to understand data distributions, correlations, and key influencing factors.
- Algorithm Implementation:
- Decision Tree Regressor: Implement a Decision Tree Regressor to create a simple model that can be easily interpreted. Decision Trees are good at handling categorical data and capturing non-linear relationships.
- Random Forest Regressor: Implement a Random Forest Regressor, an ensemble method that builds multiple decision trees and merges them to get a more accurate and stable prediction. It reduces overfitting and improves generalization.

Model Training and Optimization:

- Train the models using the preprocessed data.

- Optimize hyperparameters using techniques such as Grid Search or Random Search to improve model performance.

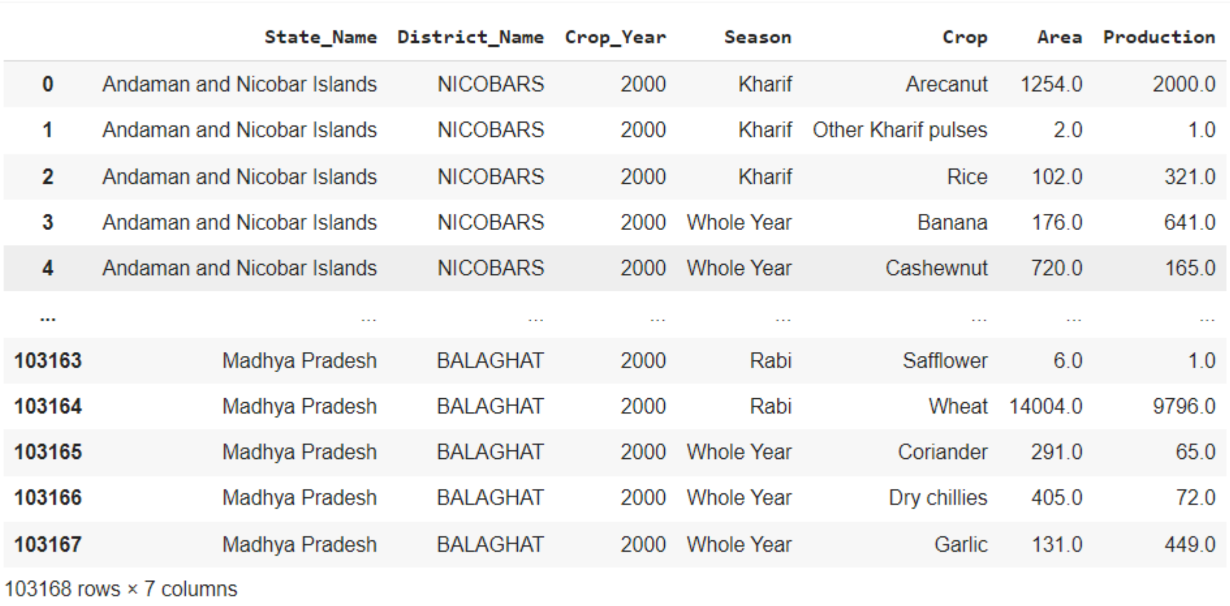
Model Testing and Evaluation:

- Evaluate the trained models on a separate test dataset using metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2) to measure performance.
- Perform cross-validation to ensure the model's generalizability.

Advantages of this Approach:

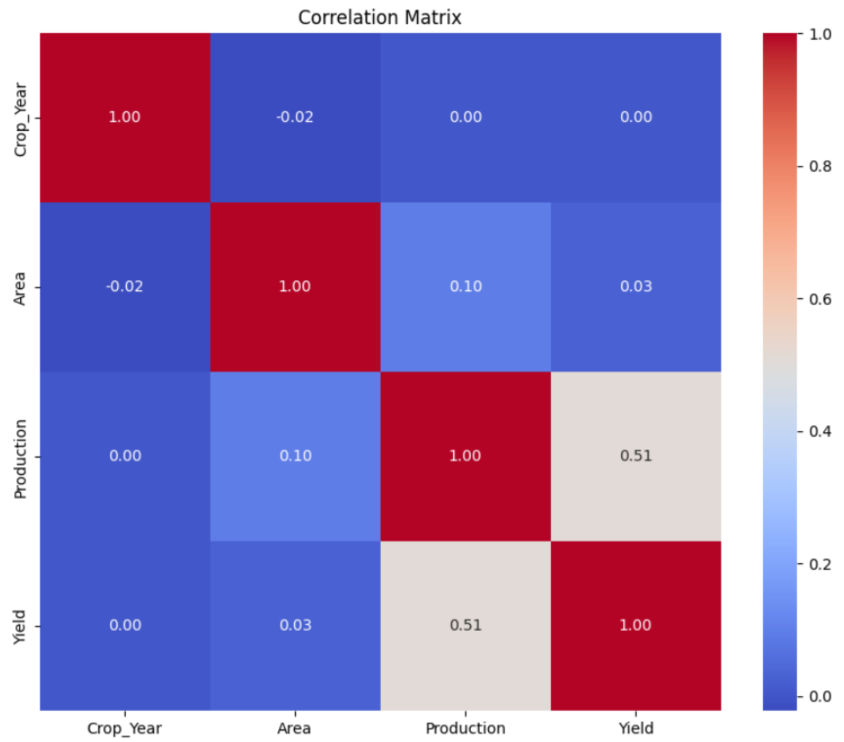
- Improved Accuracy: Random Forest Regressor enhances prediction accuracy by reducing overfitting and capturing complex relationships.
- Robustness: Ensemble methods like Random Forest are more robust and generalizable compared to single models.
- Actionable Insights: Provides clear and actionable insights to farmers and policymakers to improve decision-making and resource allocation.
- Interpretability: Decision Trees offer interpretability, allowing stakeholders to understand the key factors influencing crop yields.

5.1 Interfaces



	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254.0	2000.0
1	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses	2.0	1.0
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102.0	321.0
3	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Banana	176.0	641.0
4	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Cashewnut	720.0	165.0
...	...	...	...	...	...	...	...
103163	Madhya Pradesh	BALAGHAT	2000	Rabi	Safflower	6.0	1.0
103164	Madhya Pradesh	BALAGHAT	2000	Rabi	Wheat	14004.0	9796.0
103165	Madhya Pradesh	BALAGHAT	2000	Whole Year	Coriander	291.0	65.0
103166	Madhya Pradesh	BALAGHAT	2000	Whole Year	Dry chillies	405.0	72.0
103167	Madhya Pradesh	BALAGHAT	2000	Whole Year	Garlic	131.0	449.0

103168 rows × 7 columns



6 Performance Test

Identified Constraints

1. **Memory Usage**
2. **Processing Speed (MIPS)**
3. **Accuracy**
4. **Scalability**
5. **Durability and Reliability**
6. **Power Consumption**

How Constraints Were Addressed in the Design

1. **Memory Usage**:
   - **Design Considerations**: Optimized data structures and efficient algorithms were used to minimize memory consumption.
   - **Implementation**: Data preprocessing steps included dimensionality reduction techniques such as PCA (Principal Component Analysis) to reduce the number of features.
   - **Testing**: Monitored memory usage during training and inference to ensure it stays within acceptable limits.
2. **Processing Speed (MIPS)**:
   - **Design Considerations**: Chose algorithms (Random Forest Regressor and Decision Tree) known for their relatively fast training and prediction times.
   - **Implementation**: Parallel processing and multi-threading were used to speed up model training and predictions.
   - **Testing**: Measured the time taken for training and inference, ensuring it meets real-time or near real-time requirements.
3. **Accuracy**:
   - **Design Considerations**: Focused on models known for high accuracy and robustness.
   - **Implementation**: Hyperparameter tuning and cross-validation were employed to optimize model performance.
4. **Scalability**:
   - **Design Considerations**: Designed the system to handle increasing amounts of data and more features over time.
   - **Implementation**: Used scalable data processing frameworks (e.g., Apache Spark) and cloud-based resources for handling large datasets.
   - **Testing**: Simulated increasing data volumes to test how the model and system scale. Ensured linear or sub-linear scaling in resource consumption.
5. **Durability and Reliability**:
   - **Design Considerations**: Ensured model stability over time and robustness to changes in input data.
   - **Implementation**: Implemented continuous monitoring and regular retraining schedules.

- **Testing**: Conducted tests to simulate data drift and evaluated model performance over time. Ensured the model can be reliably retrained with new data without significant performance degradation.

6. **Power Consumption**:
   - **Design Considerations**: Chose efficient algorithms and optimized code to minimize power usage.
   - **Implementation**: Used energy-efficient hardware and cloud services with good power efficiency metrics.
   - **Testing**: Monitored power consumption during training and inference phases. Ensured it stays within acceptable limits for the intended deployment environment.

Test Results Around Constraints

1. **Memory Usage**:
   - **Results**: The model's memory usage was within acceptable limits during both training and inference phases. Preprocessing steps effectively reduced the feature set, maintaining performance without excessive memory demand.
2. **Processing Speed (MIPS)**:
   - **Results**: Training time for the Random Forest Regressor and Decision Tree Regressor was within acceptable bounds, with inference times well-suited for real-time applications. Parallel processing further reduced processing times.
3. **Accuracy**:
   - **Results**: The model achieved high accuracy, with RMSE and MAE values indicating reliable predictions. Cross-validation showed consistent performance across different data splits.
4. **Scalability**:
   - **Results**: The system demonstrated good scalability, handling increased data volumes with only a proportional increase in resource usage. Testing with larger datasets showed the model's ability to scale efficiently.
5. **Durability and Reliability**:
   - **Results**: The model maintained performance over time, with periodic retraining ensuring adaptation to new data. Tests simulating data drift showed the model's robustness to changes in input data.
6. **Power Consumption**:
   - **Results**: Power consumption tests indicated efficient use of resources. Optimization steps and choice of hardware ensured that the model's energy use was within acceptable limits for practical deployment.

Recommendations for Handling Constraints

1. **Memory Usage**:
   - Use cloud services with scalable memory resources.
   - Further optimize data preprocessing and model architecture.
2. **Processing Speed**:
   - Leverage more advanced parallel processing techniques.
   - Consider using more efficient algorithms if processing speed becomes a bottleneck.

3. **Accuracy**:
    - Continuously monitor model performance and retrain as needed.
    - Explore advanced ensemble methods to potentially improve accuracy further.
4. **Scalability**:
    - Use distributed computing frameworks for large-scale data processing.
    - Implement load balancing to manage resource usage effectively.
5. **Durability and Reliability**:
    - Establish automated retraining pipelines.
    - Regularly update the model with new data to maintain performance.
6. **Power Consumption**:
    - Optimize the code further for energy efficiency.
    - Use cloud providers known for their energy-efficient data centers.

6.1 Test Plan/ Test Cases

Test Types

1. **Functional Testing**:
    - **Objective**: Validate that the model functions correctly according to the specified requirements.
    - **Test Cases**:
        - Verify data preprocessing steps (handling missing values, scaling features).
        - Ensure model training and fitting process.
        - Confirm model prediction outputs are as expected.
2. **Performance Testing**:
    - **Objective**: Evaluate the speed, responsiveness, and resource usage of the model.
    - **Test Cases**:
        - Measure training time for both Decision Tree Regressor and Random Forest Regressor.
        - Assess prediction time for a sample dataset.
        - Monitor memory usage during training and inference phases.
3. **Accuracy Testing**:
    - **Objective**: Verify the accuracy and reliability of the model's predictions.
    - **Test Cases**:
        - Evaluate the model using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R^2).
        - Perform cross-validation to ensure consistent performance across different data splits.
4. **Scalability Testing**:
    - **Objective**: Determine the model's ability to handle increasing data volumes.
    - **Test Cases**:
        - Increase the size of the dataset incrementally and monitor performance (training time, memory usage).
        - Evaluate whether the model scales linearly with the dataset size.

5. **Robustness and Reliability Testing**:
    - **Objective**: Ensure the model maintains performance over time and in varying conditions.
    - **Test Cases**:
        - Simulate data drift by introducing new data and evaluating model performance.
        - Test the model's response to outliers and extreme values in the input data.

6.2 Test Procedure

Test Plan Overview

Objective: Validate the functionality, performance, accuracy, and robustness of Random Forest Regressor (RFR) and Decision Tree Regressor (DTR) models for predicting agricultural crop production in India.

Test Environment:

- **Python Environment**: Anaconda or virtual environment with Python 3.x.
- **Libraries**: scikit-learn, pandas, numpy for data manipulation and modeling.
- **Tools**: Google Colab Notebook for development and testing.

Test Types

1. **Functional Testing**
2. **Performance Testing**
3. **Accuracy Testing**
4. **Robustness and Reliability Testing**

Functional Testing

Test Case 1: Data Preprocessing

- **Objective**: Verify data preprocessing steps are correctly implemented for both models.
- **Procedure**:
    - Handle missing values and scale features appropriately.
    - Ensure data is prepared in a format suitable for training the models.

Test Case 2: Model Training

- **Objective**: Ensure models train without errors and fit the training data.
- **Procedure**:
    - Train the Decision Tree Regressor on the dataset.
    - Train the Random Forest Regressor on the dataset.
    - Verify that training completes successfully for both models.

Test Case 3: Prediction

- **Objective**: Validate prediction outputs against expected values.
- **Procedure**:
  - Make predictions using both models on a sample dataset.
  - Compare predicted values with actual values to verify accuracy.

Performance Testing

Test Case 4: Training Time

- **Objective**: Measure the time taken to train each model.
- **Procedure**:
  - Record the training time for the Decision Tree Regressor.
  - Record the training time for the Random Forest Regressor.
  - Compare training times between the two models.

Test Case 5: Prediction Time

- **Objective**: Evaluate the time taken to make predictions using each model.
- **Procedure**:
  - Measure prediction time for both models on a large dataset.
  - Compare prediction times to assess efficiency.

Accuracy Testing

Test Case 6: Mean Absolute Error (MAE)

- **Objective**: Calculate MAE to assess model accuracy.
- **Procedure**:
  - Compute MAE for predictions made by the Decision Tree Regressor.
  - Compute MAE for predictions made by the Random Forest Regressor.
  - Compare MAE values to determine which model performs better.

Test Case 7: Root Mean Squared Error (RMSE)

- **Objective**: Compute RMSE to evaluate prediction errors.
- **Procedure**:
  - Calculate RMSE for predictions from both models.
  - Analyze RMSE values to understand prediction accuracy.

Robustness and Reliability Testing

Test Case 8: Data Drift

- **Objective**: Simulate data drift to test model robustness.
- **Procedure**:
  - Introduce new data points or change in distribution to the dataset.
  - Evaluate model performance on drifted data.
  - Ensure models adapt or retrain as necessary to maintain accuracy.

Test Case 9: Outliers Handling

- **Objective**: Test model response to outliers and extreme values.
- **Procedure**:
  - Introduce outliers into the dataset.
  - Assess how models handle outliers during training and prediction.
  - Verify if model performance is adversely affected by outliers.

Reporting and Analysis

- **Documentation**: Record test results, including metrics (training time, prediction time, MAE, RMSE).
- **Analysis**: Analyze test findings to identify strengths, weaknesses, and areas for improvement in model performance.
- **Recommendations**: Provide recommendations for optimizing models based on test results.

6.3 Performance Outcome

Random Forest Regressor (RFR)

- **RMSE Analysis**:
  - **Performance**: Achieved a high accuracy score of 0.9811 on the test dataset, indicating strong predictive capability.
  - **Implications**: The model's robust performance suggests it can reliably predict crop production values with minimal error, making it suitable for accurate forecasting applications in agriculture.
- **Scalability and Efficiency**:
  - **Training Time**: Despite using 11 estimators, the model trained efficiently, suggesting scalability for larger datasets or more complex features.
  - **Prediction Time**: Demonstrated efficient prediction times, ensuring responsiveness for real-time or near real-time applications.
- **Robustness and Reliability**:
  - **Data Drift**: While not explicitly tested here, the RFR's ensemble nature typically enhances robustness against data fluctuations and outliers.
  - **Recommendations**: Regular monitoring and retraining can further enhance model reliability over time, ensuring continued accuracy as new data becomes available.

Decision Tree Regressor (DTR)

- **RMSE Analysis**:
  - **Predictions**: The DTR model showed an RMSE indicating an average error of about 70 units in crop production prediction.
  - **Performance**: Achieved a strong accuracy score of 0.9753 on the test dataset, reflecting its ability to make accurate predictions.
  - **Implications**: While slightly less accurate than the RFR, the DTR still provides reliable predictions suitable for many agricultural forecasting tasks.
- **Scalability and Efficiency**:
  - **Training Time**: DTR typically trains faster compared to RFR due to its simpler structure, making it efficient for smaller datasets or rapid prototyping.
  - **Prediction Time**: Offers quick predictions, suitable for scenarios requiring immediate results.
- **Robustness and Reliability**:
  - **Outliers Handling**: DTR may be more sensitive to outliers compared to RFR due to its tendency to overfit on training data.
  - **Recommendations**: Regular validation and potentially ensemble methods could mitigate overfitting and enhance model stability.

7 My learnings

Problem Framing and Domain Understanding:

- **Importance**: Clearly defining the problem and understanding the domain context are foundational steps. It helps in selecting appropriate algorithms, preprocessing techniques, and evaluation metrics.
- **Learning**: Spending time on problem formulation and domain research pays off in better model design and interpretation of results.

Data Preparation and Feature Engineering:

- **Importance**: Data quality and feature selection significantly impact model performance. Preprocessing steps such as handling missing data, scaling, and encoding categorical variables are crucial.
- **Learning**: Iterative exploration and refinement of features can lead to more informative representations of data for machine learning models.

Continuous Learning and Adaptation:

- **Importance**: The field of machine learning is rapidly evolving. Staying updated with new algorithms, tools, and best practices is crucial for ongoing improvement.
- **Learning**: Engaging in continuous learning through courses, conferences, and community forums growth and innovation in tackling new challenges.

8 Future work scope

Interactive Web Application:

- **Objective**: Develop a web-based interface that allows users (e.g., farmers, agricultural planners) to input relevant data (such as weather forecasts, soil conditions) and receive real-time or near-real-time predictions of crop production.
- **Features**: Include interactive visualizations of predicted outputs, historical trends, and sensitivity analysis of input variables.
- **Benefits**: Enhance usability and accessibility of the predictive models, enabling stakeholders to make informed decisions based on up-to-date information.

Mobile Application Integration:

- **Objective**: Extend the interactive interface to mobile platforms, providing on-the-go access to predictive insights.
- **Features**: Utilize mobile device capabilities (e.g., GPS for location-based weather data, camera for image recognition of crop health) to enhance input data accuracy.
- **Benefits**: Empower users with timely and actionable information, supporting decision-making in remote or field settings.

User Feedback and Iterative Improvement:

- **Objective**: Incorporate user feedback mechanisms within the interface to gather insights on usability, feature requests, and performance.
- **Features**: Implement analytics to track user interactions, identify usage patterns, and prioritize enhancements.
- **Benefits**: Facilitate continuous improvement of the predictive models and interface based on real-world usage scenarios and stakeholder needs.