A person holding a tablet in a field

AI-generated content may be incorrect.

**Yield Prediction Using Machine Learning**

**Table of Contents**

1. **Introduction** ........................................................................................................................... 1
2. **Related Work** ......................................................................................................................... 2
3. **Methods** .................................................................................................................................. 3

3.1 Importing Libraries, Packages, and Dataset ................................................................ 3  
3.2 Exploratory Data Analysis (EDA) .............................................................................. 3  
3.3 Checking for Outliers ................................................................................................. 4  
3.4 Data Preprocessing ...................................................................................................... 4  
3.5 Statistical Testing ........................................................................................................ 4  
3.6 Feature Engineering .................................................................................................... 5  
3.7 Feature Selection ......................................................................................................... 5  
3.8 Model Building ........................................................................................................... 5  
3.9 Model Evaluation ........................................................................................................ 6

1. **Results** .................................................................................................................................... 7
2. **Discussion** .............................................................................................................................. 9
3. **Conclusion** ............................................................................................................................ 11
4. **Contributions** ....................................................................................................................... 12
5. **References** ............................................................................................................................. 13
6. **Appendices** ............................................................................................................................ 14
7. **Introduction**

Farming is one of the most important parts of our life. It helps feed people and supports many businesses. But for many farmers, it is still hard to predict how much crop they will get from their land. This is because crop yield depends on many factors like weather, soil, water, fertilizers and more. Even though we have new technologies today, many farmers still use old methods or guesswork to plan their farming. This can lead to a lot of problems like low production, money loss or waste of resources.

In today’s world, where food demand is growing and climate conditions are changing, farmers need better tools to help them plan. A small mistake in deciding when or how to grow crops can affect not just one farm, but the entire food supply chain. That is why we wanted to use machine learning in our project to help farmers make better and smarter decisions.

The goal of our project is to build a system that can predict crop yield using data. We want to take important features like temperature, humidity, soil quality and nutrients (like nitrogen, phosphorus and potassium) and use them to predict how much crop can be produced. This can help farmers know what to expect and plan ahead.

In addition to yield, we also tried to predict the cost involved in farming. This includes how much seed is needed, the price of seeds and how big the land is. This cost prediction can help farmers understand how much money they will need and whether it is affordable for them.

At the end of this project, we also created a simple web app. This tool allows users to enter farming details and get real time predictions for both yield and cost. Our main goal was to make something helpful, easy to use, and practical for real life farming. We hope this project can make farming smarter, reduce risks and support better decision making for those who grow our food.

1. **Related Work**

This section talks about similar projects done by other researchers in the field of crop yield prediction using machine learning. Looking at other studies helps us check whether our model’s results are realistic and how our project compares with existing work.

Many researchers have tried to predict crop yield using different algorithms. For example, Jeong et al. (2016) used a Random Forest model to predict rice yield in South Korea using weather and soil data. Their model achieved an R² score of 0.89. Another study by Pantazi et al. (2016) used Neural Networks and multispectral satellite images to predict wheat yield and reported an R² of 0.82. These models showed good performance, but they mostly focused on environmental factors and did not include economic inputs like crop cost or seed usage.

In our project, we went a step further by not only using environmental features like temperature, humidity and soil quality, but also including cost-based features such as seed price and acreage. We also used feature engineering to calculate Total\_Cost, which makes our model more practical and helpful for real-life farming decisions.

Here is a comparison table of results from other studies and our project:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Study / Model Used** | **Crop Type** | **Features Used** | **Accuracy / R²** | **Notes** |
| Jeong et al. (2016) RF | Rice | Temperature, Rainfall, Soil | 0.89 | No cost-related features |
| Pantazi et al. (2016) NN | Wheat | Multispectral Images, Soil | 0.82 | Image data used |
| Our Project LightGBM | Multiple crops | Weather, Soil, NPK, Cost Features | 0.9561 | Cost + environmental features |

From this table, we can see that our model achieved a higher R² score (0.9561), showing strong performance. Unlike past research, our model also includes cost related variables, making it more useful for real world applications, especially for small scale farmers who care about both yield and expenses.

This comparison helps us understand that our model not only performs better in terms of accuracy but also offers more complete information for planning farm activities.

1. **Methods**

In this section, we explain all the steps we followed during our project, starting from collecting the dataset to building and evaluating machine learning models. Every part of the project was done carefully to make sure the model is accurate, useful and easy to understand.

* 1. **Importing Libraries, Packages, and Dataset**

We began by importing all the necessary Python libraries for data analysis and machine learning. We used pandas and numpy for data handling, matplotlib and seaborn for visualization, and scikit learn along with pycaret for building and comparing machine learning models. After setting up the environment, we imported our crop dataset into a Pandas DataFrame. The dataset included important columns like Crop\_Type, Soil\_Quality, N, P, K, Temperature, Humidity, Wind\_Speed, Soil\_pH and Acres\_Seeded. These features were later used to train and evaluate our machine learning models.

* 1. **Exploratory Data Analysis (EDA)**

After importing the dataset, we performed exploratory data analysis to better understand the structure and characteristics of the data. We used functions like .head(), .info(), and .describe() to view the first few rows, check the data types and summarize important statistics such as mean, standard deviation, minimum, and maximum values. We also created bar plots for categorical features like Crop\_Type and Soil\_Quality and distribution plots for continuous features like Temperature and Humidity. In addition, a correlation heatmap was generated to observe relationships between variables, which helped in identifying which features might be important for predicting crop yield.

* 1. **Checking for Outliers**

Outliers are unusual data points that can negatively impact the performance of machine learning models. To identify any outliers, we used box plots for key continuous variables such as N, P, K and Temperature. From the visual analysis, we did not find any significant outliers that needed to be removed. As a result, we decided to keep the complete dataset for model building.

* 1. **Data Preprocessing**

Before moving to model training, we cleaned and prepared the dataset. First, we checked for missing values using the .isna() .sum() method and confirmed that there were no missing entries. Categorical variables like Crop\_Type and Soil\_Quality were converted into numerical values using label encoding. Additionally, continuous features like Temperature, Humidity, N, P and K were normalized to ensure that all values were on a similar scale. This step helped improve the performance and training speed of the machine learning models.

* 1. **Statistical Testing**

Statistical testing was performed to verify which features had a real and significant impact on crop yield. We conducted an ANOVA test to examine whether Temperature levels and Soil Quality categories affected crop yield. In both cases, the p values were less than 0.05, indicating that they were statistically significant. Welch’s t test was used to compare yields based on small and large harvested areas and the p value of 0.036 showed that Acres\_Seeded influenced yield as well. We applied the Kruskal Wallis H test to determine if different Soil Types produced different yields and the extremely low p-value confirmed the importance of Soil\_Type. Finally, Spearman correlation was used to check the relationship between Humidity and Yield. Although the correlation was not very strong (correlation coefficient of 0.18502), it was statistically significant. These tests confirmed that Temperature, Humidity, Soil\_Quality, Soil\_Type and Acres\_Seeded were important features for building the model.

* 1. **Feature Engineering**

To make the model more realistic for real-world farming, we created three new features. The first feature was Price\_per\_unit, based on average crop selling prices collected from online agricultural sources. The second was Seeds\_per\_acre\_kg, which estimated the typical amount of seed needed to plant one acre for each crop type. Finally, we created Total\_Cost using the formula: Total\_Cost = Seeds\_per\_acre\_kg \* Price\_per\_unit \* Acres\_Seeded. These features added a financial perspective to the project, allowing us not only to predict crop yield but also to estimate the planting cost for farmers.

* 1. **Feature Selection**

After feature engineering, we carefully selected the most useful features for model building. We applied Recursive Feature Elimination (RFE) to automatically choose the best subset of features that contributed the most to model accuracy. We also referred to the correlation matrix to avoid multicollinearity and performed Chi-Square tests to verify the importance of categorical variables. Based on these methods, we finalized the features used for training as Crop\_Type, Temperature, Humidity, N, P, K and Soil\_Quality.

A screenshot of a data analysis

AI-generated content may be incorrect.

* 1. **Model Building**

Machine Learning is a branch of Artificial Intelligence that focuses on building models that learn from data and improve over time. It helps in creating systems that can predict future outcomes based on patterns found in training data. For our project, we used several machine learning algorithms to build a reliable model for crop yield prediction. The machine learning models we used are:

* LightGBM
* Random Forest Regressor
* K-Nearest Neighbors (KNN)
* XGBoost
* Gradient Boosting Regressor

These models were selected because they are popular and perform well for regression problems. As mentioned earlier, the models we used were slightly different from the ones seen in some related studies. To achieve the best possible performance, we tuned the hyperparameters for each model. This helped us find the best combination of parameters to improve model accuracy and stability. After tuning and testing, LightGBM achieved the best results and was selected for final deployment.

* 1. **Model Evaluation**

Evaluation metrics are important because they tell us how well a model is performing. They give a clear picture of the model’s strengths and weaknesses and help us decide whether the model is ready to be used in the real world. For evaluating our machine learning models, we used the following metrics:

* R² Score (Coefficient of Determination)
* Mean Absolute Error (MAE)
* Mean Squared Error (MSE)
* Root Mean Squared Error (RMSE)
* 5-Fold Cross-Validation

After training and tuning the models, we printed the evaluation metric scores for both training and testing data. We also plotted graphs to visually compare the performance of different models. All the evaluation results, including scores and plots are discussed in the Results section that follows.

1. **Results**

In this section, we explain how each machine learning model performed in predicting crop yield. We used different evaluation metrics such as R² score (which tells how accurate the model is), Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). We also did 5-Fold Cross-Validation to check if the models were consistent across different parts of the data.

After testing all models, we found that LightGBM was the most accurate and reliable model. It gave the highest R² score and the lowest error values. That means it predicted the crop yield very close to the actual values most of the time.

* 1. **LightGBM Performance Summary**

|  |  |
| --- | --- |
| **Metric** | **Value** |
| R² | 0.9561 |
| MSE | 21.1285 |
| RMSE | 4.5944 |
| MAE | 3.2862 |

These numbers show that LightGBM predicted yield values with very small errors. The MAE value (3.29) means the model was off by just around 3.29 units on average, which is very good.

* 1. **5-Fold Cross-Validation for LightGBM**

|  |  |
| --- | --- |
| **Fold** | **R² Score** |
| 1 | 0.9557 |
| 2 | 0.9546 |
| 3 | 0.9566 |
| 4 | 0.9568 |
| 5 | 0.9555 |

* Average R² Score: 0.9558
* Standard Deviation: 0.0008

We also tested LightGBM using 5-fold cross-validation to make sure it wasn’t just working well on one part of the data. Each fold gave almost the same score, and the difference between folds (standard deviation) was very small just 0.0008. This means the model is very consistent and not overfitting.

**4.3 Cost Prediction Results (Overfitting Found)**

Apart from predicting crop yield, we also tried to build a model that could predict agricultural cost. We used Random Forest, KNN, and XGBoost for this, but all three models gave a perfect score of R² : 1.0000 on training data. This is usually a sign that the model is memorizing the training data instead of learning patterns, a problem called overfitting.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **MAE** | **MSE** | **RMSE** | **R² Score** |
| Random Forest | 0.4675 | 144.64 | 5.024 | 1.0000 |
| KNN | 1.0754 | 440.81 | 12.9489 | 1.0000 |
| XGBoost | 6.2492 | 433.98 | 13.8273 | 1.0000 |

Although these results may look perfect at first, it actually means the models were not generalizing well and would probably fail on new data.

1. **Discussion**

The yield prediction model met and exceeded our expectations. Including engineered features like Total\_Cost and Seeds\_per\_acre\_kg increased model context and made the results more useful for farmers. LightGBM’s consistent performance across validation folds proved its reliability.

The cost model, however, was not deployable. It overfitted due to lack of detailed input variables such as labor cost, fuel consumption, and pest management. This aligns with similar challenges reported in literature when cost components are sparse.

We also faced deployment issues using Gradio, where changes in feature values like N, P and K had no effect on the output. After debugging, we found that the Gradio interface was loading an outdated version of the model file. Once we updated the model link and reloaded the interface, the problem was fixed.

The final web application was built and deployed using Gradio, which provided a simple and interactive user interface. Users can enter environmental and cost related values (like crop type, temperature, NPK levels and acres seeded) and get real time yield predictions. Gradio made it easy to test the model and share it without needing a backend framework like Flask.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Conclusion**

Crop yield prediction is one of the most important tasks in modern agriculture, as it helps farmers plan better and reduce uncertainty caused by weather, soil and economic factors. Our goal was to build a machine learning model that could accurately predict crop yield based on real world features such as temperature, soil nutrients, humidity and crop type.

We used several machine learning models including LightGBM, XGBoost, Random Forest, Gradient Boosting and KNN. All models were tested and compared based on performance metrics like R² score, MAE and RMSE.

After evaluating the models, we found that LightGBM performed the best. It achieved the highest accuracy (R² = 0.9561), the lowest error and showed stable results in 5-Fold Cross-Validation with very low variance. This made it the most reliable choice for predicting crop yield across different conditions.

Although we also tried building a cost prediction model, it showed signs of overfitting due to missing features such as labor and fertilizer costs. Therefore, we focused our final deployment only on yield prediction.

We concluded that LightGBM is the best model for this project, as it is accurate, consistent and works well with the features we selected. Using this model through a Gradio interface, farmers or users can easily enter values and get real time crop yield predictions. This solution can help in better decision making, reduce losses and support precision agriculture.

1. **Contributions**

|  |  |
| --- | --- |
| **Name** | **Contribution** |
| **Melbin Roy (0857356)** | * Led the yield prediction modeling process using PyCaret, evaluating multiple models like LightGBM, Random Forest, XGBoost, KNN etc. * Selected LightGBM as the best model based on R² (0.9561), RMSE, and MAE values. * Performed cross-validation and hyperparameter tuning. * Handled outlier detection, feature scaling and ensured model stability. * Finalized Gradio deployment by debugging the UI, linking the correct model and testing all parameters. * Wrote the final report sections: Results, Deployment and Model Evaluation. |
| **Arvin Jay Tambalong (0851730)** | * Worked on cost prediction modeling using Linear Regression and Random Forest for the “Total\_Cost” feature. * Added the Total\_Cost column to the standard scaler and evaluated its impact. * Helped build the Gradio interface in earlier stages and tested deployment logic. * Created the initial routing structure for the web interface and supported integration with model output. * Contributed to Discussion and Conclusion sections of the report. |
| **Steeve John Thomson (0856345)** | * Performed feature selection using Recursive Feature Elimination (RFE) and correlation filtering. * Engineered the Price\_per\_unit feature and researched crop pricing using agricultural websites. * Ran model comparisons using PyCaret and contributed to identifying the best performing regression model. * Verified the importance of selected features for yield prediction and ensured reproducibility by documenting model setup. * Contributed to Related Work and Feature Engineering sections of the report. |
| **Vishnu Mohan (0856963)** | * Worked on the cost prediction model using PyCaret, testing KNN, XGBoost, and Random Forest regressors. * Engineered and added the Seeds\_per\_acre\_kg feature based on agricultural data sources. * Analyzed cost model outputs and flagged issues with overfitting (R² : 1.0000 across models). * Used statistical testing (e.g., Welch’s t test, Spearman correlation) for hypothesis validation. * Contributed to the Methods and Statistical Analysis sections of the report. |
| **Santosh Kumar (0856971)** | * Conducted Kruskal Wallis testing to assess soil type's impact on crop yield, confirming significance (p < 0.05). * Contributed to performance diagnostics and feature tuning for the Gradient Boosting Regressor (GBR). * Assisted in UI feedback and future improvement planning for the Gradio-based web app. * Documented testing of soil related features and analyzed residual error patterns to confirm model consistency. * Supported writing of Statistical Testing and Discussion sections. |
| **Ajay Kurmaiahgari (0856346)** | * Handled feature selection using Recursive Feature Elimination (RFE), Correlation Analysis. * Contributed to parameter tuning of Gradient Boosting Regressor by adjusting max\_depth and n\_estimators. * Wrote detailed documentation for feature selection procedures to ensure reproducibility. * Supported the Methods, Feature Selection, and Hypothesis Testing sections of the report. |

1. **References**

* Jeong, J. H., et al. (2016). Random Forest based Rice Yield Prediction. Agriculture and Forest Meteorology.
* Pantazi, X. E., et al. (2016). Wheat Yield Forecasting Using Multispectral Imagery. Computers and Electronics in Agriculture.
* PyCaret Documentation. Retrieved from <https://pycaret.org>
* Scikit learn Documentation: Feature Selection Methods (RFE, Chi-square). Retrieved from <https://scikit-learn.org>
* Statistical Tutorials: ANOVA, Welch’s t test, Kruskal Wallis methods. Retrieved from <https://www.statology.org>
* Gradio Documentation: Retrieved from <https://www.gradio.app>

**9. Appendices**

* **Final Project Report**: This document contains the complete report including all sections: Introduction, Related Work, Methods, Results, Discussion, Conclusion, Contributions, and References.
* **Raw Dataset**: The original dataset containing environmental and agricultural features before any cleaning, preprocessing or feature engineering.
* **Model Training and Deployment Code (.ipynb file)**: The Jupyter Notebook file containing all code for data preprocessing, feature engineering, model training, evaluation, statistical testing and Gradio based deployment.