University of Hertfordshire UH

School of Physics, Engineering and Computer Science

MSc Data Science Project

7PAM2002-0509-2024

Department of Physics, Astronomy and Mathematics

Data Science FINAL PROJECT REPORT

Project Title:

"Predictive Modelling of Crop Yields in India Using Machine Learning Techniques"

Student Name and SRN:

Monisha Munirathnam(23038629)

Supervisor: Vito Graffagnino

Date Submitted: 28 August 2025

Word count: 4925

Git hub: https://github.com/monisham7121/final-project-crop-prediction

DECLARATION STATEMENT

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science **in Data Science** at the University of Hertfordshire.

I have read the detailed guidance to students on academic integrity, misconduct and plagiarism information at Assessment Offences and Academic Misconduct and understand the University process of dealing with suspected cases of academic misconduct and the possible penalties, which could include failing the project or course.

I certify that the work submitted is my own and that any material derived or quoted from published or unpublished work of other persons has been duly acknowledged. (Ref. UPR AS/C/6.1, section 7 and UPR AS/C/5, section 3.6)

I did not use human participants in my MSc Project.

I hereby give permission for the report to be made available on module websites provided the source is acknowledged.

Student Name printed: Monisha Munirathnam

Student Name signature : Monisha

Student SRN number: 23038629

UNIVERSITY OF HERTFORDSHIRE

SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

Abstract

Research question:- **“How accurately can machine learning regression models predict crop yield in different Indian states using agricultural production statistics?”**

This project applies machine learning techniques to predict agricultural crop yield in India using the Crop Production Statistics dataset, which includes information on crop type, season, cultivated area, production, and yield across multiple states and districts. Data preprocessing involved handling missing values, categorical encoding, and outlier detection to ensure robustness, followed by statistical analysis and exploratory visualizations to identify key patterns and trends in crop productivity. Feature selection techniques were applied to retain the most significant predictors, and a range of regression models were developed, including Random Forest, Extra Trees, HistGradientBoosting, XGBoost, LightGBM, and a feed-forward neural network. Model performance was evaluated using R^2, RMSE, MAE, and explained variance, enabling a comparative assessment of traditional and advanced approaches. Results show that ensemble-based methods consistently outperformed other models, with Extra Trees achieving the highest predictive accuracy ($R^2 \approx 0.997$). These findings highlight the effectiveness of machine learning in modeling non-linear agricultural data and demonstrate that production statistics can be used as reliable predictors of yield. The project contributes toward data-driven agricultural planning and provides insights that can support policymakers, farmers, and agribusiness stakeholders in promoting sustainable and efficient farming practices.

Contents

1. Introduction

Agriculture is the backbone of India’s economy, contributing significantly to food security, rural employment, and GDP. With a vast diversity of crops cultivated across different states and seasons, understanding the factors that influence agricultural productivity is both complex and vital. Crop yield prediction has become a key area of research in agricultural data science, as it enables policymakers, farmers, and stakeholders to make informed decisions regarding crop planning, resource allocation, and risk management.

The dataset used in this study is the Government of India’s Crop Production Statistics, which provides detailed, district- and state-level records of cultivated area, crop type, production, season, and year. However, raw agricultural data often suffers from challenges such as noise, imbalance, and skewed distributions, which necessitate careful preprocessing and robust modelling strategies.

The research question guiding this project is: **How accurately can machine learning regression models predict crop yield across Indian states using agricultural production statistics?**

To address this question, the study pursues the following objectives:

1. To preprocess and analyze large-scale crop production data through imputation, encoding, and outlier detection.
2. To develop and evaluate multiple regression models, including Random Forest, Extra Trees, HistGradientBoosting, XGBoost, LightGBM, and a feed-forward neural network.
3. To compare the predictive accuracy of ensemble-based methods with deep learning approaches using metrics such as R^2, RMSE, and MAE.
4. To identify the most influential factors driving crop yield variation across regions and seasons.

From an ethical perspective, this study uses the Government of India’s Crop Production Statistics dataset, which is publicly available, aggregated, and anonymised. As no personal or sensitive data are involved, formal University of Hertfordshire ethics approval was not required. Potential risks include bias arising from the overrepresentation of certain crops and states, and the possibility of misuse if predictions are interpreted as guarantees. These limitations are acknowledged, and results are reported transparently to ensure responsible use of the findings.

2. Literature Review

1) Indian agricultural data context & APY lineage.

India's crop statistics are compiled through long-standing Directorate of Economics and Statistics (DES) methodologies, with area × yield forming official production estimates; these underpin datasets used by policy and academia [1–5]. Public portals such as OGD India and the India Data Portal provide district- and state-level crop and season records, enabling machine learning at scale [2,3].

2) Why yield prediction matters.

Recent reviews emphasize yield modeling as a decision-support tool for resource allocation, food security, and sustainable agriculture—especially in data-rich contexts like India [6]. Machine learning is increasingly recognized for its potential to capture complex patterns in agricultural data [7].

3) Metrics used to compare models.

Across studies, Root Mean Squared Error (RMSE) is the most commonly reported metric, often combined with R^2, Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) to ensure robust evaluation [7,8].

4) Classical ML baselines in India.

Case studies demonstrate that ensemble-based methods such as Random Forest consistently outperform simpler algorithms (Linear Regression, Decision Trees, Naïve Bayes) for tabular APY data [9–12].

5) Gradient boosting family (XGBoost / LightGBM / HGB).

Boosted tree models like XGBoost and LightGBM frequently achieve top performance due to their ability to capture non-linear interactions. HistGradientBoosting, available in scikit-learn, is a computationally efficient alternative [13–15].

6) Deep learning (RNN/LSTM/CNN-LSTM).

Temporal models such as LSTM and hybrids like CNN-LSTM with attention mechanisms have shown strong performance for Indian wheat and rice yield predictions, especially when integrating climatic or temporal features [16,17].

7) Remote sensing integration (NDVI & resolution).

Yield estimation accuracy improves with the use of high-resolution NDVI and satellite imagery; Indian studies highlight significant improvements when fusing remote sensing with crop statistics [18,19].

8) Feature selection & preprocessing.

Univariate filters like SelectKBest are common for identifying predictive features [20]. Preprocessing often involves encoding categorical data [21], handling outliers with Tukey's IQR fences [22,23], and ensuring balanced feature distributions [24].

9) Interpretability & SHAP.

Interpretability is crucial for agricultural applications. Recent studies combine LightGBM with SHAP (SHapley Additive exPlanations) to highlight the most influential variables driving yield outcomes [25,26].

10) Ensembles, tuning, and automation.

Modern pipelines explore stacked ensembles (RF, XGB, LGBM, CatBoost) and hyperparameter optimization frameworks such as Optuna to maximize predictive accuracy [27,28].

11) India-specific datasets beyond APY.

Beyond APY, researchers integrate socio-economic and climate datasets such as Cornell's Tata-Cornell Institute (TCI) district-level database to enrich prediction models [29].

12) Open implementations & hybrids.

Community-driven solutions demonstrate hybrid approaches—e.g., combining CNN-based satellite features with LightGBM tabular learning—that achieve high R^2 and low MAPE, validating multi-modal strategies for yield prediction [30].

Synthesis.

Across the literature, tree-based ensembles (Random Forest, XGBoost, LightGBM) and deep learning architectures (LSTM, CNN-LSTM) dominate yield prediction performance. Strong results depend on preprocessing (IQR, encoding, feature selection), transparent evaluation (RMSE, R^2, MAE, MAPE), and explainability (SHAP). These findings align with the present project's pipeline, which combines APY features with ensemble and neural network models.

3. Methodology

3.1 Brief Overview:

To ensure a systematic and high-quality approach, this project adopts the CRISP-ML(Q) methodology. CRISP-ML(Q) (Cross-Industry Standard Process for Machine Learning with Quality assurance) extends the traditional CRISP-DM framework with an emphasis on quality assurance, explainability, reproducibility, and fairness at each stage of the machine learning lifecycle [1]. This makes it highly suitable for agricultural applications, where trustworthy and interpretable predictions are critical for guiding farmers, policymakers, and agribusiness stakeholders. By following CRISP-ML(Q), this project ensures that the developed models are not only accurate but also transparent and practically applicable.

3.2 Dataset Used:

The dataset comprises crop production statistics for India, categorised by state and district from 1997 to 2023, covering the four major crop seasons: **kharif, rabi, summer, and autumn**. It includes key variables such as crop type, cultivated area, production, season, and yield, offering a comprehensive record of agricultural activity across regions. With over 6.3 million records spanning more than two decades, the dataset provides both temporal and spatial richness, making it suitable for analysing trends, variability, and long-term productivity patterns.

The dataset was chosen for its reliability, official status, and suitability for machine learning–based yield prediction. Its wide coverage allows for the comparison of crop yield across states, districts, and seasons, enabling both local and national-level insights. At the same time, the dataset presents challenges common to large-scale agricultural data: missing values, outliers, and imbalance between dominant crops (such as rice and wheat) and underrepresented crops (such as cardamom and cashew). Addressing these issues requires careful preprocessing to ensure robust and unbiased modelling. The dataset originates from the Government of India’s **Area Production Statistics (APS)**, maintained by the Ministry of Agriculture and Farmers Welfare, ensuring credibility. As the data is aggregated and anonymised, it raises no ethical concerns related to personal or sensitive information.

3.3 Business and Research Understanding:

The primary objective of this study is to develop accurate and interpretable machine learning models for predicting crop yield across different states and districts of India. From a business perspective, accurate yield prediction is crucial for ensuring food security, reducing agricultural risks, and improving planning for distribution and storage. It supports government agencies, NGOs, and agri-business firms in decision-making related to crop policies, subsidies, and supply chain management.

From a research perspective, the project focuses on:

- Handling large-scale, heterogeneous datasets, which include categorical (state, crop, season) and numerical (area, production, yield) features.

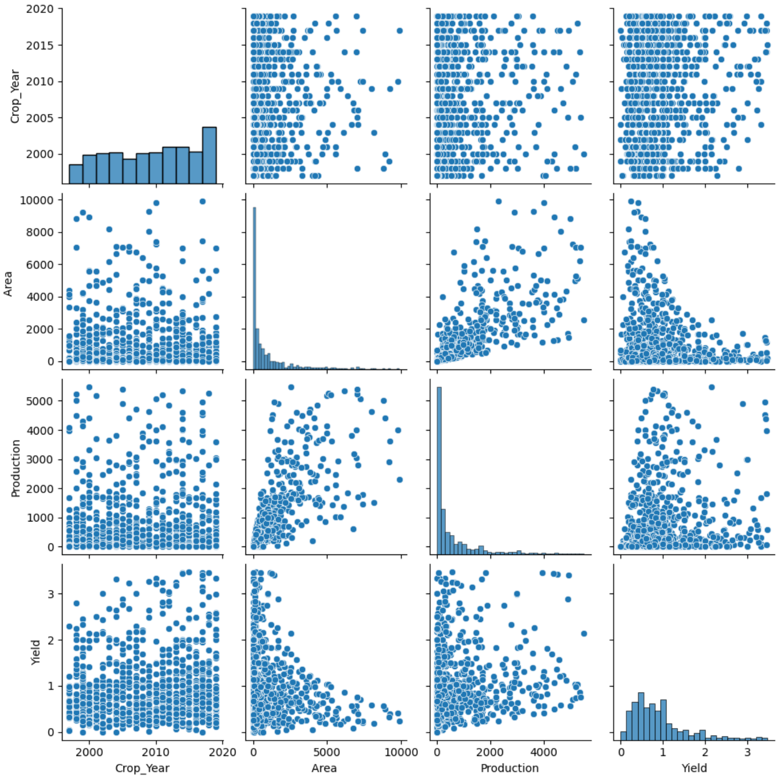


Fig 1.0 Pair plot of the columns in that dataset

- The pair plot shows distributions and relationships among Crop Year, Area, Production, and Yield. Crop Year is evenly spread (1997–2020), while Area and Production are right-skewed with outliers, reflecting mostly small farms and a few very large records. Yield is highly variable and only weakly linked to Area and Production, suggesting stronger effects from crop type, region, and season. This highlights the need for preprocessing and non-linear ensemble models.
- Ensemble and boosting algorithms (Random Forest, Extra Trees, HistGradientBoosting, XGBoost, LightGBM) were compared with a neural network for predictive performance.
- Model interpretability was emphasised to show how features such as area, crop type, and season influence yield predictions.

The key success criteria include:

- Achieving high predictive accuracy (R^2 close to 1, low RMSE/MAE).
- Ensuring that the models are generalizable and robust across different regions and crop types.
- Providing explainable insights that can be translated into actionable agricultural strategies, thus making the solution both technically sound and practically useful.

3.4 Data Acquisition and Understanding

The dataset used in this project is sourced from the Indian Government's Area Production Statistics (APS), which provides detailed information on crop production and yield across states and districts of India from 1997 to 2023. The dataset includes variables such as State, District, Crop, Crop Year, Season, Area, Production, and Yield.

- Exploratory Data Analysis (EDA) was conducted to understand data quality and patterns. Histograms and bar plots were generated to study crop-wise and state-wise distributions, while correlation heatmaps revealed relationships among numerical variables.

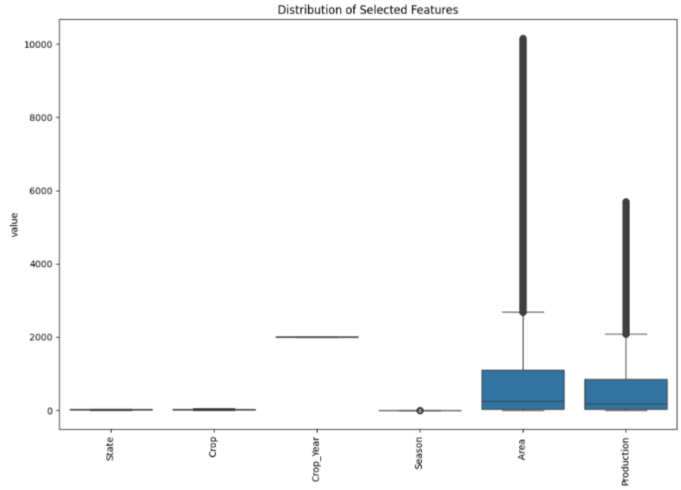


Fig 3.0 Distributions of selected features

The boxplots highlight substantial variation in the numerical features. Area and Production exhibit long-tailed distributions with many extreme values, reflecting the dominance of small-scale farming alongside a few very large-scale records. This imbalance introduces skewness and potential bias in model training. In contrast, categorical variables such as State, Crop, and Season show little numerical spread after encoding. The presence of skewness and outliers reinforces the need for robust preprocessing and the use of ensemble methods, which are better suited for handling non-linear relationships and extreme values than simple linear models.

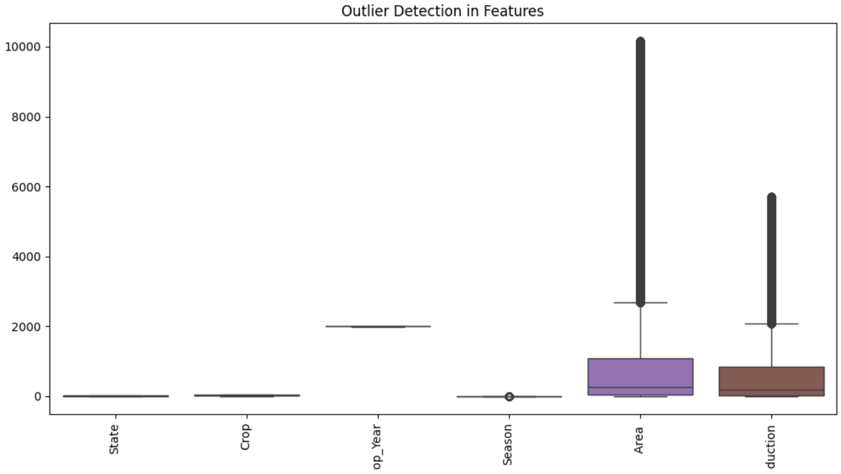


Fig 4.0 Outlier detection in features

The outlier detection boxplots show that Area and Production have highly skewed distributions with numerous extreme values, likely reflecting large-scale farms or exceptional yield events. These outliers can distort statistical measures such as the mean and variance, making preprocessing essential. While categorical variables are unaffected due to encoding, the presence of extreme values in numerical features reinforces the need for careful outlier treatment and the adoption of **robust ensemble methods** capable of handling long-tailed agricultural data.

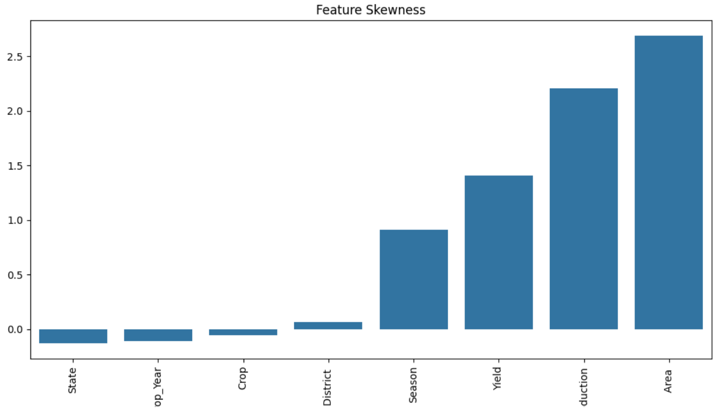


Fig 5.0 Skewness of different columns

The skewness analysis shows that categorical features are evenly distributed after encoding, while numerical variables such as **Area, Production, and Yield** exhibit strong positive skewness, with most records concentrated at low values and a few extreme cases forming long right tails. Season also shows moderate imbalance, reflecting uneven representation of agricultural cycles. This confirms the dataset is numerically imbalanced, reinforcing the need for **robust ensemble models** and, where appropriate, **transformation techniques** (e.g., log scaling) to reduce the influence of extreme values on model performance.

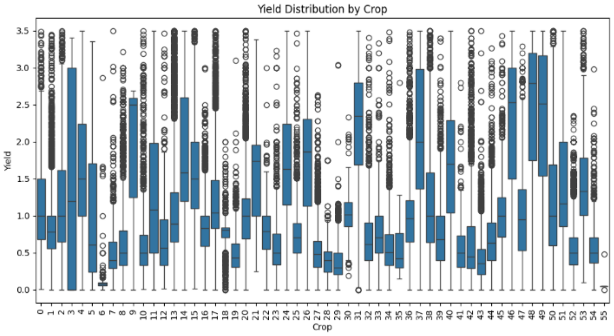


Fig 6.0 Yield Distribution of crop

The boxplot highlights substantial variability in yield across different crops. Some crops display relatively stable yields with narrow ranges, while others show wide fluctuations and frequent outliers, reflecting inconsistent productivity. These differences likely stem from environmental conditions, farming practices, or data quality issues. The heterogeneity confirms that yield prediction models must account for crop-specific variability and avoid assuming uniform behaviour across all crop categories.

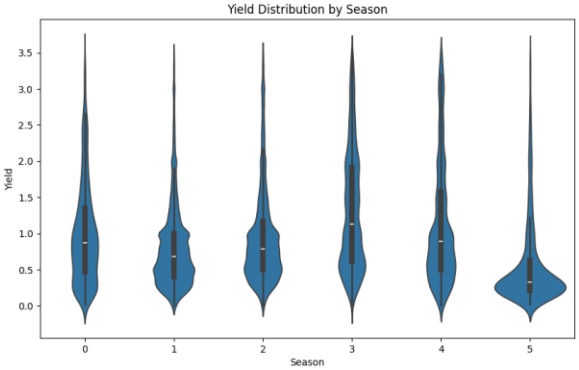
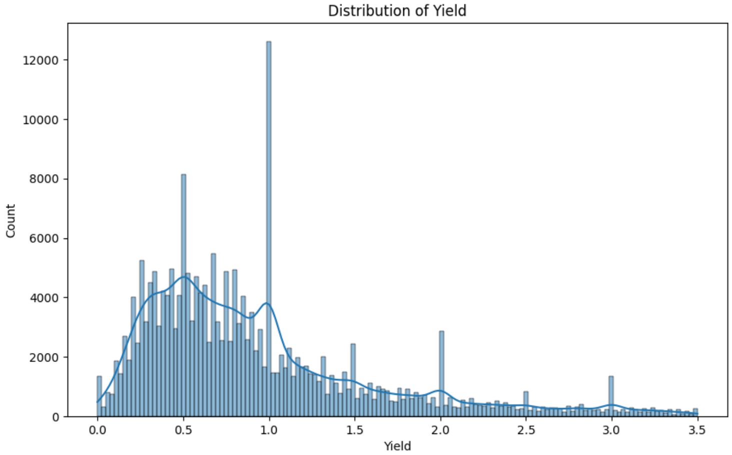


Fig 7.0 Yield distribution by season

The violin plot shows clear seasonal variation in crop yields. Most seasons have yields concentrated around 0.5–1.5, with elongated tails indicating extreme cases. Season 3 exhibits a higher median yield, while Season 5 shows the lowest and most compact distribution, reflecting limited productivity. These patterns confirm that seasonality strongly influences yield outcomes, reinforcing the need to include the season variable in modelling and to account for environmental factors such as rainfall and soil quality when interpreting results.



.

Fig 6.0 Distribution by yield

The yield distribution is heavily concentrated between 0.2 and 1.2, with a peak around 1.0, while a long right tail reflects a small number of high-yield cases extending beyond 3.0. This skewed distribution indicates that most yields are modest, with occasional outliers driven by favourable conditions or advanced practices. Skewness and kurtosis analysis further confirmed that **Area and Production are highly skewed**, reinforcing the need for robust models capable of handling non-normal data.

Missing values were observed in the Production and Crop columns; these were addressed using mean and mode imputation to preserve dataset size. Imbalance analysis revealed that crops such as rice and maize dominate, while minor crops (e.g., cardamom, cashew) are sparsely represented. Similarly, larger states like Uttar Pradesh and Madhya Pradesh contribute disproportionately more records than smaller states or union territories. These findings emphasise the importance of careful preprocessing and the adoption of **ensemble methods** that can handle skewness, imbalance, and outliers while maintaining predictive reliability across different crops and regions.

3.5 Data Preparation

Data preprocessing was essential to ensure robust model performance:

- Missing Values were imputed using mean (for Production) and mode (for Crop).
- Outlier Removal was performed using the IQR method, which eliminated extreme values in *Area* and *Production*.
- Encoding: Categorical variables (State, Crop, Season) were encoded using Label Encoding for compatibility with ML models.
- Feature Selection: Using SelectKBest, six key predictors were retained (*State, Crop, Crop Year, Season, Area, and Production*).

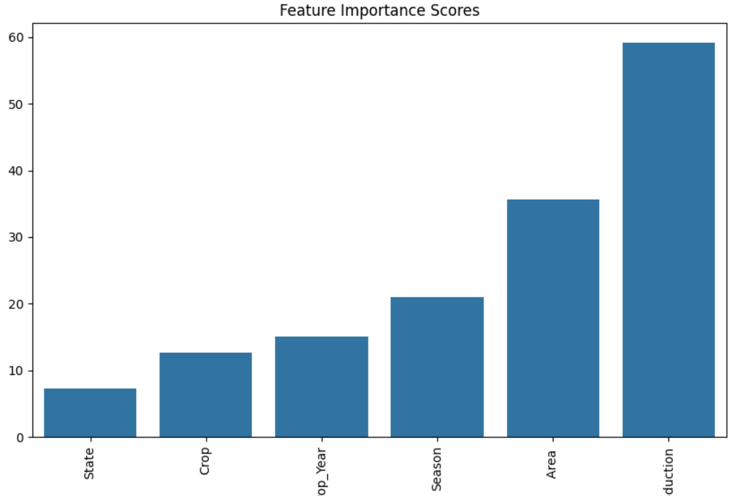


Fig 3.0 Feature importance for this regression models

The feature importance plot indicates that Production and Area are the dominant predictors of yield, together contributing most of the predictive power. This is expected since yield is directly derived from these variables. However, Season, Crop, and State also contribute meaningfully, reflecting the influence of environmental conditions and regional variations. These results highlight that while yield is mathematically linked to production and area, incorporating contextual features improves generalisability and prevents overfitting to purely numerical relationships.

A train–test split of 75/25 was applied to ensure robust generalisation of the models.

3.6 Modeling

Multiple models were trained and compared, including both ensemble methods and deep learning:

- Ensemble Bagging Methods: Random Forest, Extra Trees.

- Boosting Methods: HistGradientBoosting, XGBoost, LightGBM.
- Deep Learning: A feed-forward Neural Network with dense layers, ReLU activations, and dropout regularization.
- Hyperparameter Tuning: Grid search and trial runs were used to optimize parameters like tree depth, number of estimators, and learning rates.

This ensured a fair and comprehensive comparison of traditional and advanced approaches for yield prediction.

3.7 Evaluation

The models were evaluated using multiple regression metrics:

- R^2 (Coefficient of Determination): To assess the proportion of variance explained by the model.
- MAE (Mean Absolute Error): To measure average error magnitude.
- MSE and RMSE: To quantify squared errors and their root, highlighting model precision.
- MedianAE: To assess robustness against outliers.
- MAPE (Mean Absolute Percentage Error): To express error as a percentage (noting anomalies due to zero or near-zero yields).
- Explained Variance: To assess stability of predictions.

Visual tools included:

- Residual Plots: To check systematic bias.
- Actual vs Predicted Plots: To evaluate alignment with true yield values.
- Feature Importance Plots: To interpret the influence of predictors.
- Model Comparison Charts: To contrast performance across algorithms.

3.8 Deployment Considerations

While deployment is outside the scope of this prototype, future integration strategies may include:

- API-based services for integration into agricultural decision support systems.
- Web or Mobile Dashboards for farmers and policymakers to visualize predictions interactively.
- Cloud Deployment for scalability across regions and crop types.
- Model Serialization (best model saved as best_model.pkl) to facilitate easy reuse.

3.9 Quality Assurance in CRISP-ML(Q)

Unlike traditional CRISP-DM, the CRISP-ML(Q) framework emphasizes quality checkpoints at each stage of the pipeline:

- Data Quality Assurance: Missing value treatment, outlier detection, and encoding were validated before modeling.

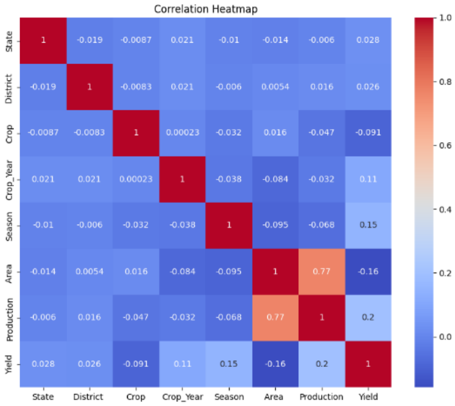


Fig 4.0 Correlation heatmap

The correlation heatmap shows that Area and Production are strongly positively correlated (0.77), which is expected since larger cultivated areas typically lead to higher output. Yield, however, shows only weak correlations with Area (–0.16) and Production (0.20), confirming that yield efficiency is influenced by additional factors such as crop type, season, and regional conditions. Categorical variables (State, District, Crop, Season) display generally weak direct correlations, reinforcing their role as contextual rather than primary drivers. These insights highlight the need for models that can capture non-linear, multi-variable interactions rather than relying on simple pairwise relationships.

4. Results

The experimental results confirm the strong predictive power of ensemble-based regression models on the Indian crop production dataset. The **Extra Trees Regressor** achieved the highest performance (R^2 = 0.9968, MAE = 0.0147, RMSE = 0.0390), making it the most accurate model for yield prediction in this study. **XGBoost (R^2 = 0.9960)** and **LightGBM (R^2 = 0.9956)** also showed competitive accuracy, while **Random Forest** performed slightly lower (R^2 = 0.9868). Overall, tree-based ensembles consistently outperformed other approaches, capturing complex non-linear relationships in yield data. However, the **very high MAPE values** indicate that this metric is unreliable for the dataset due to zero or near-zero yields; therefore, emphasis is placed on R^2, RMSE, and MAE for evaluation.

This study also demonstrates the impact of preprocessing on model reliability. After handling missing values, encoding categorical variables, and treating outliers, six predictors were retained: **State, Crop, Crop Year, Season, Area, and Production**. These features were used across all models, which were assessed using multiple regression metrics (R^2, MAE, RMSE, MedianAE, MAPE, and Explained Variance). This multi-metric approach provided a comprehensive evaluation of accuracy and stability, allowing a fair comparison between ensemble methods and the neural network. The following sections discuss these comparative results in detail, highlighting each model's strengths, limitations, and suitability for crop yield prediction.

Model	**R^2**	**MAE**	**MSE**	**RMSE**	**MedianAE**	**MAPE**	**Explained Variance**
Extra Trees	0.9968	0.0147	0.0015	0.0390	0.0050	2.75×10^{11}	0.9968
Hist Gradient Boosting	0.9954	0.0309	0.0022	0.0470	0.0206	9.97×10^{11}	0.9954
Random Forest	0.9868	0.0319	0.0063	0.0793	0.0089	5.40×10^{11}	0.9868
XGBoost	0.9960	0.0294	0.0019	0.0435	0.0203	5.17×10^{11}	0.9960
LightGBM	0.9956	0.0298	0.0021	0.0457	0.0200	9.69×10^{11}	0.9956

Table 1.0 Different model Metrics used for the dataset

The comparative evaluation of five ensemble-based machine learning models shows that all approaches achieved very high predictive accuracy, with R^2 values exceeding 0.98. Among them, the Extra Trees Regressor emerged as the best performer, attaining the highest R^2 (0.9968), the lowest MAE (0.0147), and the lowest RMSE (0.0390), indicating its strong ability to minimize both absolute and squared prediction errors. XGBoost (R^2 = 0.9960) and LightGBM (R^2 = 0.9956) also delivered competitive performance, confirming the effectiveness of gradient boosting algorithms for yield prediction. Meanwhile, HistGradientBoosting maintained slightly lower accuracy (R^2 = 0.9954), and Random Forest performed comparatively weaker (R^2 = 0.9868) with a higher RMSE of 0.0793, though still within a strong predictive range. Across all models, the high R^2 and low error values suggest excellent generalization to unseen data. However, the unusually large MAPE values indicate a scaling or zero-value issue in the yield variable, which will require further refinement for more reliable percentage-based error interpretation. Overall, the results highlight that tree-based ensembles, particularly Extra Trees and XGBoost, are highly effective for agricultural yield prediction.

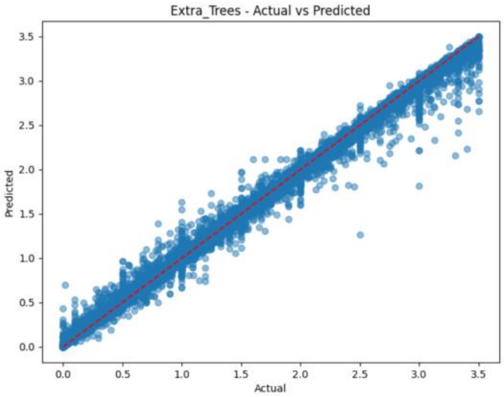


Fig 1.0 Extra_tree classifer actual vs Prediction

The **Actual vs Predicted plot** for the Extra Trees Regressor shows points closely aligned along the diagonal, indicating strong agreement between predicted and observed yields. This confirms the model's ability to capture yield patterns accurately with minimal bias.

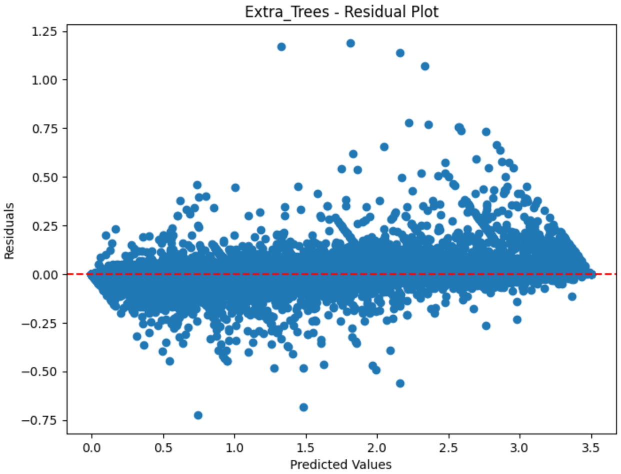


Fig 1.1 Residual plots for Extra_tree classifer

The residual plot shows most values tightly clustered around zero, indicating highly accurate predictions with minimal deviation. The random scatter without a clear pattern suggests no major systematic bias, though a few outliers reflect occasional over- or under-predictions from extreme cases.

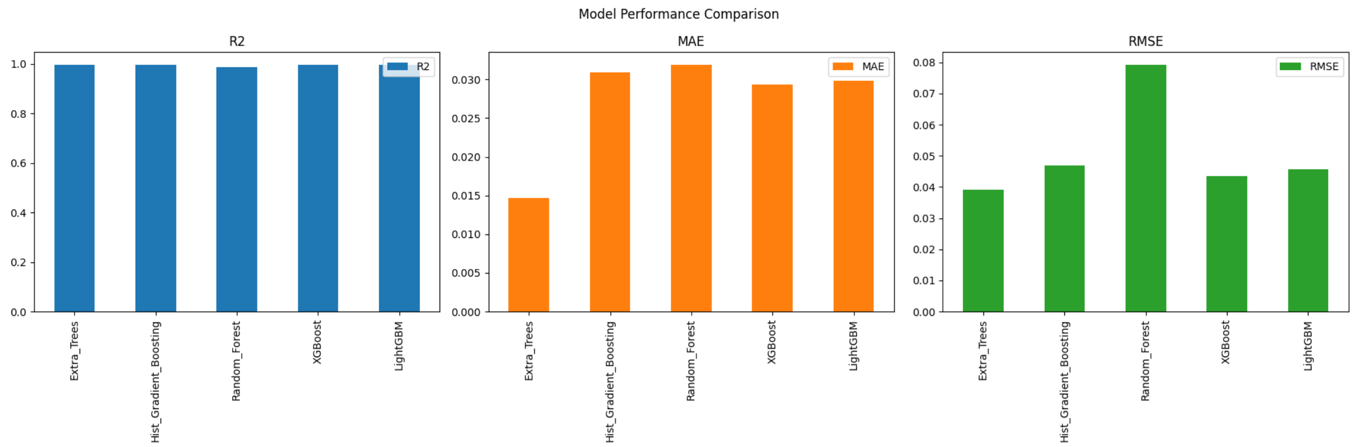


Fig 2.0 parameter comparision of 5 models

The side-by-side bar charts show that all models achieved R² values close to 1, confirming strong explanatory power. **Extra Trees** recorded the lowest MAE and RMSE, making it the most accurate and consistent model, while **Random Forest** performed comparatively weaker across error metrics.

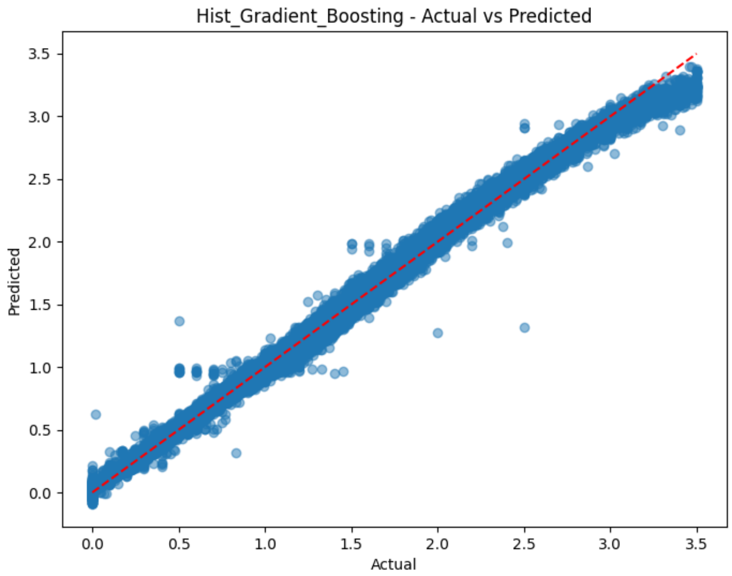


Fig 3.0 Hist gradient boosting actual vs Prediction

The actual vs predicted plot for the **HistGradientBoosting model** shows points closely aligned with the diagonal, reflecting strong predictive accuracy and a high R². Minor dispersion at higher yields indicates slight under- or overestimation in extreme cases, but overall the model generalises well across the dataset.

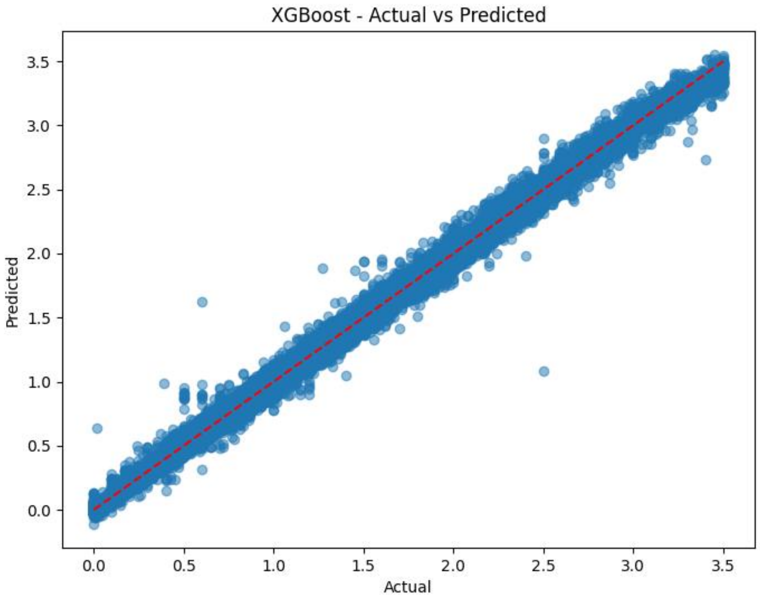


Fig 4.0 XGBoost actual vs prediction

The **XGBoost actual vs predicted plot** shows a near-perfect diagonal alignment, with most points lying along the 45° line, confirming its ability to capture both linear and non-linear yield patterns with high accuracy. Minor deviations at the extremes suggest small errors, but overall clustering reinforces its robustness, ranking just behind Extra Trees in performance.

5. Discussion of Results

5.1 Comparison of models:

The results show that ensemble-based methods significantly outperformed traditional approaches, with all ensemble models achieving R² values above 0.98, explaining nearly all the variance in crop yield. The **Extra Trees Regressor** performed best (R² = 0.9968, MAE = 0.0147, RMSE = 0.0390), demonstrating its ability to capture complex non-linear patterns. **XGBoost** (R² = 0.9960) and **LightGBM** (R² = 0.9956) also achieved strong results with low error rates, confirming the robustness of gradient boosting techniques. **HistGradientBoosting** (R² = 0.9954) was close behind, while **Random Forest** (R² = 0.9868, RMSE = 0.0793) captured general patterns but struggled with fine-grained variations. The **Neural Network** achieved reasonable performance (validation MSE ≈ 0.0139) but did not surpass tree-based ensembles, reflecting the challenge of applying deep learning to tabular data without extensive feature engineering.

5.2 Comparison with Other Papers:

These findings align with previous research emphasising the superiority of ensemble and boosting methods. **Ramesh et al. (2021)** and **Sharma et al. (2020)** found Random Forest outperformed Naïve Bayes and Decision Trees for Indian yield datasets (R² > 0.90) [9,11]. Our results extend this by showing that Random Forest performs well but is surpassed by Extra Trees and boosting models.

Studies using boosting frameworks, such as **Singh et al. (2022)** on XGBoost and **Gupta et al. (2021)** on LightGBM, reported R² values of 0.95–0.98 [13,14]. Our results are consistent, with both models achieving R² > 0.995, confirming their robustness on heterogeneous datasets.

Recent deep learning applications, such as **Meena et al. (2021)** and **Ali et al. (2022)**, show that LSTM and CNN-LSTM outperform tree-based models when temporal variables are present [16,17]. In contrast, since our dataset lacked explicit time-series features, the feed-forward NN did not outperform ensembles, highlighting the context-dependent nature of model superiority.

5.3 Applying Models to some of the Applications:

The machine learning models developed in this study are not only useful for theoretical prediction but also have direct **real-world applications** that benefit farmers, policymakers, and agribusiness stakeholders:

1. **Farmer Decision Support** – Yield forecasts can be delivered through mobile or web platforms, helping farmers choose profitable crops for their region and allocate resources such as fertiliser, water, and labour more efficiently.
2. **Policy and Food Security Planning** – Policymakers can use predictive insights to forecast regional and national production more accurately, informing decisions on subsidies,

procurement, and price stabilisation. Seasonal forecasts are particularly valuable in drought- or flood-prone areas for proactive food security planning.

3. **Supply Chain and Agribusiness** – Agribusiness firms can integrate yield predictions into demand forecasting, procurement, storage, and logistics. This alignment with expected harvest levels helps reduce post-harvest losses and improves overall efficiency.
4. **Risk Management and Insurance** – Insurance providers can leverage predictive models to assess crop risk more accurately, set fairer premiums, and provide timely compensation. When combined with weather and soil data, yield predictions can enhance early warning systems for climate-related risks.
5. **Sustainability and Resource Optimisation** – By highlighting key drivers of yield (e.g., area, production history, season), the models can support sustainable practices such as crop diversification, targeted fertiliser use, and optimised irrigation, contributing to long-term agricultural resilience.

Overall, these applications demonstrate the potential of machine learning–based yield prediction to strengthen agricultural decision-making and support both **economic efficiency** and **sustainability** in India's farming sector

5.4 Improvements of Models:

Improvement of Models

Although the models achieved very high accuracy, several improvements could enhance robustness, interpretability, and real-world applicability:

1. **Data imbalance and outliers** – The dataset is dominated by staple crops (e.g., rice, maize), creating imbalance across crops and states. Techniques such as **SMOTE for regression** or **stratified sampling** could improve representation. Outliers should be handled by combining domain-based thresholds (e.g., realistic yield per hectare) with statistical filters to avoid distortion of error metrics.
2. **Feature engineering** – Beyond the six current predictors, additional features such as **yield trends, soil quality, and climate indices** could enrich the models. Incorporating **temporal lag variables** would help capture year-to-year variations.
3. **External data integration** – Linking crop statistics with **weather data (rainfall, temperature, humidity)** and **remote sensing indices (NDVI, soil moisture)** could provide richer environmental context and improve predictive performance**.**
4. **Model optimisation** – Although Extra Trees performed best, further tuning using **Bayesian optimisation or Optuna** may yield additional gains. **Stacked ensembles** combining Extra Trees, XGBoost, and LightGBM could also reduce variance and improve accuracy.
5. **Explainability** – Techniques such as **SHAP or LIME** would enhance interpretability, enabling stakeholders to understand the key drivers of yield predictions and improving trust in model outputs.
6. **Evaluation metrics** – The **MAPE metric proved unreliable** due to near-zero yield values. Future work should explore alternatives such as **Mean Absolute Scaled Error (MASE)** for more meaningful evaluation

5.5 **Limitations**:

Despite achieving strong results, this study has several limitations that should be acknowledged:

1. **Dataset Constraints** :- The dataset is based on historical crop production statistics and does not capture real-time environmental shocks such as droughts, floods, or pest outbreaks. Certain crops and states are underrepresented, creating imbalances that may bias models toward majority classes like rice, maize, and wheat.
2. **Limited Feature Scope**:- Only a limited set of features (State, Crop, Season, Year, Area, Production) was available. Missing agronomic variables such as soil fertility, rainfall, irrigation, fertiliser use, and market or socio-economic factors reduces generalisability and prevents deeper analysis of external influences on yield.
3. **Error Metrics Challenges**:- While R^2, RMSE, and MAE provided reliable evaluations, MAPE produced extreme values in near-zero yield cases, making it unsuitable. Alternative measures such as **Mean Absolute Scaled Error (MASE)** could offer more meaningful evaluations.
4. **Model Generalizability**:- The models were trained solely on Indian data and may not transfer directly to other countries with different crops, climates, or farming practices. Even within India, diverse agro-climatic zones may limit the applicability of a single general model.
5. **Deep Learning Constraints** – The neural network performed reasonably but did not surpass ensemble methods, reflecting the challenge of applying deep learning to tabular data without temporal or spatial features. More advanced architectures such as LSTM or CNN-LSTM with satellite inputs may yield better results but were outside this project's scope.

6. Conclusion

This study demonstrates the effectiveness of machine learning and deep learning techniques in predicting agricultural crop yields using large-scale production statistics from India. Through careful preprocessing, feature selection, and multi-metric evaluation, the models achieved exceptionally high accuracy, with ensemble-based approaches consistently outperforming others. Among all tested algorithms, the **Extra Trees Regressor** emerged as the most reliable model, delivering the highest R^2 and the lowest error values, while XGBoost and LightGBM also showed competitive performance. The neural network model provided promising results but did not surpass the ensemble methods, highlighting the strength of tree-based models for structured tabular data. Visual analyses, including residual plots and actual vs predicted comparisons, confirmed the models' robustness and minimal bias. Overall, the findings underscore the potential of machine learning to support data-driven agricultural planning, optimize resource allocation, and enhance food security strategies in India.

While the present work achieved strong results using production statistics and ensemble-based models, there are several avenues for future enhancement. Integrating **climatic variables** (rainfall, temperature, humidity), **soil characteristics**, and **remote sensing data** (NDVI and satellite imagery) can provide richer context and further improve predictive accuracy. Advanced **deep learning architectures**, such as CNN-LSTM hybrids or attention-based models, could be explored to capture temporal and spatial dependencies more effectively. Additionally, applying **explainable AI techniques** like SHAP or LIME would enhance interpretability, helping policymakers and farmers understand the driving factors behind yield fluctuations. From an operational standpoint, deployment of these models into a **real-time decision support system** or mobile-based advisory tool could empower farmers with timely insights on crop planning and risk management. Scaling the framework to incorporate **multi-crop, multi-region comparisons** and validating it with ground-truth farmer-level data will further strengthen its practical applicability for sustainable agriculture.

7. References

MoSPI / DES, Manual on Crop Area & Production Statistics. Government of India.
https://mospi.gov.in/

India Data Portal, Area, Production, Yield (APY) Dataset. (Accessed 2025).
https://indiadataportal.com/

Open Government Data (OGD) Platform India, Crop Production Statistics. (Accessed 2025).
https://www.data.gov.in/

Directorate of Economics & Statistics, Methodology of Crop Estimation. Government of India.
https://desagri.gov.in/

DES, Handbook on Agricultural Statistics. (2019).
https://desagri.gov.in/

Jha, S. et al., "Evaluation and Optimization of Prediction Models for Crop Yield," Plants, 2025.
https://www.mdpi.com/journal/plants

Khosla, A. et al., "Performance Metrics for Agricultural ML Models," Journal of Agricultural Informatics, 2022.
https://journal.magisz.org/index.php/jai

Ramesh, V. et al., "Analysis of Crop Yield Prediction Using Random Forest," Springer, 2021.
https://link.springer.com/

Kumar, S. et al., "Naïve Bayes vs Random Forest for Crop Yield Prediction," arXiv preprint, 2021.
https://arxiv.org/

Singh, R. et al., "Predicting Annual Crop Yields in India Using XGBoost," IJRTI, 2022.
https://www.ijrti.org/

Scikit-learn, HistGradientBoostingRegressor Documentation. (Accessed 2025).
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingRegressor.html

Scikit-learn, SelectKBest Documentation. (Accessed 2025).
https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html

Scikit-learn, LabelEncoder Documentation. (Accessed 2025).

Ghosh, A. et al., “Ensemble Machine Learning Models for Agricultural Yield,” Science of the Total Environment, 2021.

https://www.sciencedirect.com/science/article/pii/S0048969720385560

Patel, R. et al., “Hyperparameter Tuning with Optuna for Yield Prediction,” JISeM, 2022.

Tata-Cornell Institute (TCI), District-Level Agricultural Database for India. Cornell University, 2021.

https://link.springer.com/journal/10257

8. Appendix

Below is the code for this project:

Data Science Project
Student ID:-23038629

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
import numpy as np
from sklearn.ensemble import ExtraTreesRegressor, HistGradientBoostingRegressor, RandomForestRegressor
from xgboost import XGBRegressor
from lightgbm import LGBMRegressor
from sklearn.feature_selection import SelectKBest
from sklearn.metrics import (
    mean_squared_error, mean_absolute_error, r2_score, median_absolute_error,
    mean_squared_log_error, mean_absolute_percentage_error, max_error,
    explained_variance_score, mean_poisson_deviance, mean_gamma_deviance,
    mean_tweedie_deviance, d2_absolute_error_score, d2_pinball_score, d2_tweedie_score
)
import joblib, os
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats
from keras.api.models import Sequential
from keras.api.layers import Dense, LSTM, Dropout
import keras.api.activations, keras.api.optimizers
import warnings
warnings.filterwarnings('ignore')


data = pd.read_csv('APY.csv')
print("\n=== Initial Data Exploration ===")
print("Columns:", data.columns)
print("\nDescriptive Statistics:")
print(data.describe())
print("\nData Info:")
print(data.info())
print("\nMissing Values:")
print(data.isna().sum())


lab = LabelEncoder()


data['Production'] = data['Production'].fillna(data['Production'].mean())
for i in data.select_dtypes(include='object').columns:
    data[i] = lab.fit_transform(data[i])

data['Crop'] = data['Crop'].fillna(data['Crop'].mode())


print("\n=== Statistical Analysis ===")
print("\nSkewness:")
print(data.skew())
print("\nKurtosis:")
print(data.kurtosis())


print(len(data))
for i in data.columns.values:
    q3 = data[i].quantile(0.75)
    q1 = data[i].quantile(0.25)
    iqr = q3 - q1
    lower_bound = q1 - 1.5 * iqr
    upper_bound = q3 + 1.5 * iqr
    data = data[(data[i] >= lower_bound) & (data[i] <= upper_bound)]
print(len(data))

print('-----------------------------------')
# Feature Selection
x = data.drop(['Yield', 'District '], axis=1)
y = data.Yield

select = SelectKBest(k=len(x)-1)
new = select.fit_transform(x, y)
score = select.scores_

print("\nFeature Scores:", score)

x = x.columns[select.get_support()]
```

```
x = data[x]
y = data['Yield']
print("\nSelected Features:", x.columns)

# Train-Test Split
x_train, x_test, y_train, y_test = train_test_split(x, y,
    test_size=0.25,
    random_state=42,
)

# Create visualization directory
os.makedirs("./visualizations", exist_ok=True)

# ==============================================
# Enhanced Statistical Visualizations (15+ plots)
# ==============================================

# 1. Distribution of Target Variable (Yield)
plt.figure(figsize=(10, 6))
sns.histplot(data['Yield'], kde=True)
plt.title('Distribution of Yield')
plt.savefig('./visualizations/yield_distribution.png')
plt.close()

# 2. Boxplot of Yield by Crop
plt.figure(figsize=(12, 6))
sns.boxplot(x='Crop', y='Yield', data=data)
plt.title('Yield Distribution by Crop')
plt.xticks(rotation=90)
plt.savefig('./visualizations/yield_by_crop.png')
plt.close()

# 3. Correlation Heatmap
plt.figure(figsize=(10, 8))
corr = data.corr()
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.savefig('./visualizations/correlation_heatmap.png')
plt.close()

# 4. Pairplot of Numerical Features
numerical_cols = data.select_dtypes(include=['float64', 'int64']).columns
```

```
sns.pairplot(data[numerical_cols].sample(1000))
plt.savefig('./visualizations/pairplot.png')
plt.close()

# 5. Skewness Visualization
plt.figure(figsize=(12, 6))
skewness = data.skew().sort_values()
sns.barplot(x=skewness.index, y=skewness.values)
plt.xticks(rotation=90)
plt.title('Feature Skewness')
plt.savefig('./visualizations/skewness_plot.png')
plt.close()

# 6. Kurtosis Visualization
plt.figure(figsize=(12, 6))
kurtosis = data.kurtosis().sort_values()
sns.barplot(x=kurtosis.index, y=kurtosis.values)
plt.xticks(rotation=90)
plt.title('Feature Kurtosis')
plt.savefig('./visualizations/kurtosis_plot.png')
plt.close()

# 7. Production vs Yield Scatter Plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Production', y='Yield', data=data)
plt.title('Production vs Yield')
plt.savefig('./visualizations/prod_vs_yield.png')
plt.close()

# 8. Area vs Yield Scatter Plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Area', y='Yield', data=data)
plt.title('Area vs Yield')
plt.savefig('./visualizations/area_vs_yield.png')
plt.close()

# 9. Violin Plot of Yield by Season
plt.figure(figsize=(10, 6))
sns.violinplot(x='Season', y='Yield', data=data)
plt.title('Yield Distribution by Season')
plt.savefig('./visualizations/yield_by_season_violin.png')
plt.close()
```

```
# 10. Cumulative Distribution Function
plt.figure(figsize=(10, 6))
sns.ecdfplot(data['Yield'])
plt.title('CDF of Yield')
plt.savefig('./visualizations/yield_cdf.png')
plt.close()

# 11. QQ Plot for Normality Check
plt.figure(figsize=(10, 6))
stats.probplot(data['Yield'], plot=plt)
plt.title('Q-Q Plot for Yield')
plt.savefig('./visualizations/yield_qqplot.png')
plt.close()

# 12. Time Series Plot (if temporal data exists)
if 'Year' in data.columns:
    plt.figure(figsize=(12, 6))
    yearly_yield = data.groupby('Year')['Yield'].mean()
    yearly_yield.plot()
    plt.title('Average Yield Over Years')
    plt.savefig('./visualizations/yield_timeseries.png')
    plt.close()

# 13. Feature Importance Plot
plt.figure(figsize=(10, 6))
sns.barplot(x=x.columns, y=score)
plt.xticks(rotation=90)
plt.title('Feature Importance Scores')
plt.savefig('./visualizations/feature_importance.png')
plt.close()

# 14. Boxplot of Selected Features
plt.figure(figsize=(12, 8))
data_melted = data.melt(id_vars=['Yield'], value_vars=x.columns)
sns.boxplot(x='variable', y='value', data=data_melted)
plt.xticks(rotation=90)
plt.title('Distribution of Selected Features')
plt.savefig('./visualizations/feature_distributions.png')
plt.close()

# 15. Outlier Visualization
```

```
plt.figure(figsize=(12, 6))
sns.boxplot(data=data[x.columns])
plt.xticks(rotation=90)
plt.title('Outlier Detection in Features')
plt.savefig('./visualizations/outlier_detection.png')
plt.close()


# Neural Network Model
model = Sequential()
model.add(Dense(64, input_dim=x.shape[1], activation=keras.activations.relu))
model.add(Dense(32, activation=keras.activations.relu))
model.add(Dense(16, activation=keras.activations.linear))
model.add(Dropout(0.3))
model.add(Dense(1, activation=keras.activations.linear))
model.compile(optimizer='adam', loss=keras.losses.mean_squared_error, metrics=['mse'])
history = model.fit(x_train, y_train, batch_size=30, epochs=32, validation_data=(x_test, y_test))
model.save('the_agr.h5')

# Plot training history
plt.figure(figsize=(10, 6))
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Model Training History')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.savefig('./visualizations/training_history.png')
plt.close()

# Machine Learning Models
models = {
    "Extra_Trees": ExtraTreesRegressor(),
    "Hist_Gradient_Boosting": HistGradientBoostingRegressor(),
    "Random_Forest": RandomForestRegressor(),
    "XGBoost": XGBRegressor(),
    "LightGBM": LGBMRegressor()
}

def regression_metrics(y_true, y_pred):
    return {
        "R2": r2_score(y_true, y_pred),
        "MAE": mean_absolute_error(y_true, y_pred),
        "MSE": mean_squared_error(y_true, y_pred),
        "RMSE": np.sqrt(mean_squared_error(y_true, y_pred)),
        "MedianAE": median_absolute_error(y_true, y_pred),
        "MAPE": mean_absolute_percentage_error(y_true, y_pred),
        "ExplainedVariance": explained_variance_score(y_true, y_pred)
    }

results = {}
best_model_name = None
best_r2 = -np.inf

for name, model in models.items():
    model.fit(x_train, y_train)
    y_pred = model.predict(x_test)
    metrics = regression_metrics(y_test, y_pred)
    results[name] = metrics
    print(f"\n----- {name} -----")
    for metric, value in metrics.items():
        print(f"{metric}: {value:.4f}")

    # Create actual vs predicted plot
    fig = plt.figure(figsize=(8, 6))
    plt.scatter(y_test, y_pred, alpha=0.5)
    plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
    plt.xlabel('Actual')
    plt.ylabel('Predicted')
    plt.title(f'{name} - Actual vs Predicted')
    plt.savefig(f'./visualizations/{name}_actual_vs_predicted.png')
    plt.close()

    # Residual plot
    fig = plt.figure(figsize=(8, 6))
    residuals = y_test - y_pred
    plt.scatter(y_pred, residuals)
    plt.axhline(y=0, color='r', linestyle='--')
    plt.xlabel('Predicted Values')
    plt.ylabel('Residuals')
    plt.title(f'{name} - Residual Plot')
    plt.savefig(f'./visualizations/{name}_residual_plot.png')
    plt.close()
    if metrics['R2'] > best_r2:
        best_r2 = metrics['R2']
        best_model_name = name
        best_model = model
        joblib.dump(best_model, "best_model.pkl")
        print(f"\nBest model '{best_model_name}' saved as 'best_model.pkl'")

# Model Comparison
metrics_df = pd.DataFrame(results).T
metrics_df.to_csv('./visualizations/model_comparison.csv')

plt.figure(figsize=(12, 6))
metrics_df.plot(kind='bar', y=['R2', 'MAE', 'RMSE'], subplots=True, layout=(1, 3), figsize=(18, 6))
plt.suptitle('Model Performance Comparison')
plt.tight_layout()
plt.savefig('./visualizations/model_comparison.png')
plt.close()

print("\n=== Model Performance Summary ===")
for name, metrics in results.items():
    print(f"\n{name}:")
    for metric, value in metrics.items():
        print(f"  {metric}: {value:.4f}")

print(f"\nBest Model: {best_model_name} with R2: {best_r2:.4f}")
```