International Journal of Science, Strategic Management and Technology
Volume **02** Issue **05** May-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

Crop Yield Prediction using Machine Learning

Sharukesh M, Naveen Kumar P, Dr. SK. Piramu Preethika

Department of Computer Science and Information Technology, School of Computing Sciences,

https://doi.org/10.55041/ijsmt.v2i5.112

Cite this Article: M, S. & P, N. K. (2026). Crop Yield Prediction using Machine Learning. International Journal of Science, Strategic Management and Technology, *02*(05). https://doi.org/10.55041/ijsmt.v2i5.112

Abstract:
Predicting crop yield is a critical challenge in modern agriculture due to the complex interplay of environmental, soil, and climatic variables. Traditional empirical methods have become increasingly unreliable as rapid environmental changes unfold. This paper presents a machine learning-based framework for crop yield prediction employing the Support Vector Machine (SVM) algorithm as the primary classifier. The study incorporates advanced data preprocessing, including feature selection methods such as Recursive Feature Elimination (RFE) and Modified Recursive Feature Elimination (MRFE), alongside data-balancing techniques including SMOTE and ROSE to address class imbalance. Comparative analysis against Naïve Bayes and Decision Tree baselines demonstrates that SVM achieves superior predictive accuracy. A Flask-based web application was developed to deliver real-time predictions from environmental input parameters. Results confirm that the proposed approach significantly improves prediction reliability, offering a scalable decision-support tool for farmers navigating volatile climatic conditions.
Keywords: Crop yield prediction; Support Vector Machine; Feature selection; SMOTE; Precision agriculture; Machinelearning;Python;Flask.

1. INTRODUCTION

Agriculture remains the backbone of global food security, yet the sector faces mounting pressure from climate variability, population growth, and resource constraints. Accurate prediction of crop yields is indispensable for supply chain management, policy planning, and optimising farm-level resource allocation. Historically, yield estimation relied on expert agronomic knowledge and empirical rules derived from decades of observation. However, the accelerating pace of climate change has rendered such heuristics insufficiently adaptive.

Machine learning (ML) offers a data-driven alternative that can capture non-linear relationships among environmental variables without requiring explicit mechanistic modelling. Among the many ML paradigms investigated for agricultural applications, Support Vector Machines (SVM) have emerged as particularly attractive because of their strong theoretical foundations in statistical learning theory, robustness to high-dimensional feature spaces, and capacity to generalise well from modest training sets.

This paper proposes an end-to-end crop yield prediction system that integrates data preprocessing, feature selection, class-balancing, SVM-based classification, and a lightweight deployment layer. Contributions include: (i) a systematic comparison of SVM against Naïve Bayes and Decision Tree classifiers, (ii) application of SMOTE, ROSE, and MWMOTE oversampling to address class imbalance, (iii) evaluation of RFE and MRFE for feature subset selection, and (iv) deployment of the trained model via a Flask REST API, enabling real-time predictions.

2. PROBLEM STATEMENT AND MOTIVATION

Crop cultivation decisions—what to plant, how much to irrigate, and when to harvest—are governed by a confluence of biotic factors (pest incidence, crop variety) and abiotic factors (rainfall, temperature, humidity, soil pH, and macronutrient levels). Traditional decision support either relies on static lookup tables or univariate regression models that fail to capture multivariate interactions.

Existing ML approaches, such as Decision Tree ensembles (e.g., Random Forest), have shown promise but are susceptible to overfitting when irrelevant features inflate the input dimensionality. Furthermore, many publicly available agricultural datasets suffer from severe class imbalance: certain crops or yield brackets are under-represented, leading to biased classifiers that excel only on majority classes.

The present work addresses these gaps by (a) applying rigorous feature selection before model training, (b) correcting class imbalance via oversampling, and (c) benchmarking multiple algorithms to identify the most accurate predictor. The ultimate objective is to equip farmers with a reliable, easy-to-use prediction tool accessible through a web browser.

3. SYSTEM DESIGN AND METHODOLOGY

3.1 Dataset and Environmental Variables

The model ingests seven input features widely reported in the agronomic literature as the primary determinants of crop yield: soil nitrogen content (N), phosphorous (P), potassium (K), ambient temperature (°C), relative humidity (%), soil pH, and rainfall (mm). Target labels correspond to recommended crop categories derived from historical yield data collected across diverse Indian agro-climatic zones.

3.2 Data Preprocessing Pipeline

Raw sensor and archival data exhibit common data quality issues. The preprocessing pipeline addresses these in four sequential stages:

- Missing value imputation using column-wise median substitution to preserve distributional symmetry.
- Outlier detection via the Interquartile Range (IQR) method; extreme values beyond 1.5×IQR are clipped rather than discarded to retain sample size.
- Min-max normalisation applied to all continuous features, projecting values into [0, 1] to ensure equitable contribution to the kernel computation.
- Label encoding of the categorical crop target variable.

3.3 Feature Selection

Recursive Feature Elimination (RFE) was applied with SVM as the estimator to rank features by their contribution to prediction accuracy. Modified Recursive Feature Elimination (MRFE) further refines this by iteratively evaluating candidate feature subsets under cross-validation, retaining only those features whose removal causes a statistically significant drop in performance ($p < 0.05$). This dual approach prevents redundant features from inflating training time and overfitting risk.

3.4 Class Imbalance Handling

Three oversampling strategies were evaluated: (i) ROSE (Random Over-Sampling Examples) generates synthetic samples via bootstrap resampling in feature space; (ii) SMOTE (Synthetic Minority Over-sampling Technique) interpolates between minority-class nearest neighbours; and (iii) MWMOTE (Majority Weighted Minority Over- sampling Technique) concentrates synthetic generation near the decision boundary where misclassification probability is highest. Cross-validation results showed that SMOTE yielded the most balanced improvement in minority-class recall without excessive precision loss.

3.5 Classification Models

Three classifiers were benchmarked. **Support Vector Machine (SVM):** An RBF-kernel SVM with regularisation parameter C = 10 and kernel coefficient $\gamma = 0.1$, optimised via grid search. SVM maximises the margin between decision boundaries, making it inherently resistant to outliers and noise. **Naïve Bayes:** A probabilistic baseline that assumes conditional independence among features. **Decision Tree:** A hierarchical rule-learner susceptible to overfitting, included as a second baseline.

3.6 Data Flow

Raw input → Missing value imputation → Outlier clipping → Feature selection (RFE/MRFE) → SMOTE oversampling → Train/test split (80:20) → SVM training → Serialisation (pickle) → Flask REST endpoint → Web UI prediction display.

4. SYSTEM REQUIREMENTS

4.1 Non-Functional Requirements

Performance: The system must produce predictions within two seconds for single-sample inference on standard consumer hardware. Scalability: The backend must support concurrent requests from multiple users during peak farming seasons. Reliability: Prediction accuracy must not degrade below 85% under dataset drift conditions within a 12-month operational window. Security: All data transmissions must use HTTPS; input validation must prevent adversarial inputs from corrupting the model state. Usability: The web interface must be operable without prior ML knowledge; input fields are self-descriptive and include validation ranges.

4.2 Functional Requirements

The system collects environmental parameters through a web form, preprocesses them through the same pipeline used during training, invokes the serialised SVM model, and returns a predicted crop label with an associated confidence estimate. Integration hooks are provided for IoT sensor APIs (ZigBee, LoRa, Z-Wave) to automate data ingestion from smart farming devices.

4.3 Software and Hardware Stack

Software: Python 3.10, scikit-learn 1.3, Flask 3.0, NumPy 1.26, pandas 2.1, Anaconda Navigator (development environment), SQLite 3 (lightweight logging). Hardware (minimum): Intel Core i5, 8 GB RAM, 1 TB HDD, 15-inch display; 64-bit Windows 10 or Linux.

5. IMPLEMENTATION

5.1 Model Training

Model training was performed inside a Jupyter Notebook environment within Anaconda Navigator. The dataset was split 80:20 into training and test partitions using stratified sampling to preserve class proportions. Hyperparameter tuning employed 5-fold cross-validated grid search over $C \in \{0.1, 1, 10, 100\}$ and $\gamma \in \{0.001, 0.01, 0.1, 1\}$. The final model was serialised using Python's pickle module for deployment.

5.2 Flask Web Application

The Flask application exposes two routes: GET / renders the input form (index.html); POST /predict accepts seven numeric parameters (N, P, K, temperature, humidity, pH, rainfall), constructs a NumPy array, invokes model.predict(), and renders result.html with the predicted crop label. The

application runs in production mode (debug=False) and can be containerised for cloud deployment using Docker or hosted on a low-cost VPS to provide remote access to farmers.

5.3 Anaconda Environment

All dependencies are managed within a dedicated conda virtual environment, ensuring reproducibility. The Anaconda Navigator GUI provides additional tools including JupyterLab for iterative development, Spyder for debugging, and VSCode integration for version- controlled code management. SQLite was used to log prediction requests with input parameters and timestamps to support post-hoc model drift analysis.

6. RESULTS AND EVALUATION

6.1 Unit and Integration Testing

Unit testing validated each preprocessing function in isolation, verifying correct handling of edge cases such as NaN inputs, values outside the valid feature range, and single-class batches. Integration testing confirmed that the Flask endpoint correctly serialised model inputs, invoked inference, and returned predictions across all crop classes. A regression test suite was implemented to detect performance degradation following model updates.

6.2 Classifier Performance Comparison

Table 1 summarises performance on the held-out test set after applying SMOTE oversampling and RFE feature selection:

Classifier	Accuracy	Precision	F1-Score
SVM (proposed)	94.3%	93.8%	94.1%
Naïve Bayes	82.1%	81.4%	81.7%
Decision Tree	87.6%	86.9%	87.2%

Table1.Classificationperformanceontheheld-outtestset(n= 20%ofdataset).

SVM outperforms both baselines by a margin of at least 7 percentage points in accuracy. The improvement is most pronounced for minority crop classes, confirming that the combined effect of SMOTE oversampling and RFE feature selection successfully mitigates both class imbalance and the curse of dimensionality.

6.3 Effect of Feature Selection

Without feature selection, SVM accuracy on raw data was 89.2%. Applying RFE reduced the input dimension from 7 to 5 features (excluding correlated soil ion proxies) and improved accuracy to 92.4%. MRFE further pruned to 4 features with accuracy reaching 94.3%, confirming that removing correlated attributes sharpens the decision boundary.

7. CONCLUSION AND FUTURE WORK

This paper demonstrated that SVM-based crop yield prediction, augmented with systematic feature selection and class-balancing, achieves state-of-the-art accuracy (94.3%) on real-world agricultural datasets. The Flask deployment layer makes the system accessible to non-expert stakeholders, bridging the gap between algorithmic sophistication and practical farm- level application.

Future directions include: (i) incorporating deep learning architectures (LSTM) to model temporal dynamics of multi-season yield trends; (ii) integrating satellite remote-sensing imagery as additional input features; (iii) deploying IoT-connected sensors (ZigBee, LoRa) for automated real-time data ingestion; (iv) extending the framework to support hyperlocal predictions at the field
-parcel level; and (v) federated learning to preserve data privacy when aggregating readings from multiple farms.

REFERENCES

[1] Liakos, K. G., Busato, P., Moshou, D., Pearson, S., & Bochtis, D. (2018). Machine learning in agriculture: A review. Sensors, 18(8), 2674.

[2] Chlingaryan, A., Sukkarieh, S., & Whelan, B. (2018). Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture. Computers and Electronics in Agriculture, 151, 61–69.

[3] Pantazi, X. E., Moshou, D., Alexandridis, T., Whetton, R. L., & Mouazen, A. M. (2016). Wheat yield prediction using machine learning and advanced sensing techniques. Computers and Electronics in Agriculture, 121, 57–65.

[4] Jeong, J. H., Resop, J. P., Mueller, N. D., Fleisher, D. H., Yun, K., Butler, E. E., ... & Kim, S. H. (2016). Random forests for global and regional crop yield predictions. PloS One, 11(6), e0156571.

[5] Vapnik, V. N. (1995). The Nature of Statistical Learning Theory. Springer, New York.

[6] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321–357.

[7] Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of Machine Learning Research, 3, 1157–1182.

[8] Van der Laan, M. J., Polley, E. C., & Hubbard, A. E. (2007). Super learner. Statistical Applications in Genetics and Molecular Biology, 6(1).