International Journal of Computer Techniques–IJCT Volume 12 Issue 6, November 2025
Open Access and Peer Review Journal ISSN 2394-2231 https://ijctjournal.org/

A Machine Learning Approach for Forecasting Vegetable Prices in Indian APMC Markets Using XGBoost

Mr. Umesh Nanavare, Mr. Shantanu Patil, Mr. Om Patil, Mr. Sunny Khokle, Mr. Harpalsing Rajput
Department of Computer Science and Engineering
MIT Art, Design & Technology University, Pune – 412201, India
Email: {patilshantanu260, ompatil0251, sunny.khokle, harpalsingrajput007}@gmail.com

***Abstract*—Agricultural price volatility of perishable commodities is a persistent driver of rural distress in India. This work develops an extensive machine-learning pipeline for short-term forecasting of modal prices for vegetables (cabbage, cauliflower, and green chilli) in Pune district APMC markets (2023–2025). We design a robust feature engineering suite (lagged, rolling, Fourier-based seasonal encodings), formulate the XGBoost learning objective with regularization rigorously, and deploy an operational inference API. Extensive experiments compare XGBoost against ARIMA and LSTM baselines across multiple performance metrics, showing 25–40% improvement in MAE and RMSE. The paper discusses interpretability, deployment concerns, and policy implications for integrating forecasts into market decision systems. Code artifacts and model metadata are structured for reproducibility and productionization.**

***Index Terms*—Agricultural forecasting, APMC markets, XGBoost, time-series forecasting, feature engineering, model deployment.**

I. Introduction

INDIA's agricultural markets are characterized by heterogenous supply chains, regional demand patterns, and pronounced seasonality. Perishable vegetables, in particular, exhibit rapid price swings driven by harvest timing, rainfall, and short-term transportation constraints. These dynamics pose substantial risk to farmers' incomes and make market planning difficult for traders and policymakers.

APMC (Agricultural Produce Market Committee) mandis are central to India's wholesale trade. While they provide important price and quantity information, their usage for forecasting is limited. Traditional statistical techniques—ARIMA/SARIMA and exponential smoothing—capture linear temporal structure but fail under abrupt shocks and nonlinear interactions across variables. The rise of accessible datasets (notably Agmarknet) and increased computational resources enables deploying machine-learning models for short-term price forecasting that are both accurate and operationally viable.

This paper focuses on a deployable, interpretable approach using eXtreme Gradient Boosting (XGBoost) for short-term forecasting of vegetable modal prices. XGBoost is selected for its ability to handle heterogeneous tabular features, built-in regularization, and interpretability via feature importance measures. We aim to provide a full pipeline: data ingestion, cleaning, feature engineering, model training & selection, deployment, and monitoring. The contributions are:

- A reproducible end-to-end forecasting pipeline using XGBoost tailored to APMC datasets.
- A formal mathematical derivation of the boosting objective and regularization.
- An extensive empirical evaluation on Pune district vegetable markets (2023–2025) comparing XGBoost to ARIMA and LSTM baselines.
- A practical deployment architecture, including a REST API and monitoring strategy, aimed at supporting integration into governmental and commercial dashboards.

II. Background and Motivation

Farmers' incomes are acutely sensitive to price volatility in perishable crops. Given low storage capacity and high perishability, even short-term price drops can cause severe losses. Forecasts that provide reliable short-term estimates (1–7 days horizon) can significantly alter harvesting and selling decisions, enabling:

- Timely storage or delayed sale to capture better prices.
- Optimized transportation routing to markets with higher expected prices.
- Policy interventions, such as buffer supplies, to stabilize consumer prices.

From a data availability perspective, the Agmarknet portal provides daily modal price data across APMC markets. While these data are useful, they contain a mix of noise, missing dates, and reporting anomalies. A robust pipeline must therefore address data quality and carefully engineer temporal features that capture seasonality and momentum.

III. Literature Survey

This section provides a deeper review of relevant literature, grouped into classical statistical models, machine learning models, and deep learning / hybrid approaches.

A. Classical Statistical Models

Time-series analysis has a long tradition in commodity price forecasting. ARIMA models (Box-Jenkins methodology) capture autoregressive and moving-average components and can be extended to seasonal ARIMA (SARIMA) to handle periodicity [1]. However, ARIMA presumes stationary residuals after differencing and struggles with nonlinear dependencies and exogenous shocks. The inefficacy of ARIMA in volatile short-term agricultural forecasting is documented in multiple applied studies, where MAE and RMSE increase notably during high-variance periods.

B. Machine-learning Methods

Tree-based ensembles—Random Forests (RF) and Gradient Boosted Decision Trees (GBDT)—provide non-linear modeling capacities on tabular data. XGBoost [3] extends GBDT with computational optimizations and regularization. Empirical results across domains show XGBoost's robustness, and in agricultural contexts it consistently outperforms linear baselines. Paul et al. [2] reported improved MAE using RF and GBM for brinjal forecasting in Odisha, reinforcing ensemble efficacy in crop price tasks.

C. Deep Learning and Hybrid Models

Deep learning, particularly LSTM and Transformer variants, learns long-term dependencies and is useful when multi-variate high-frequency data (including exogenous weather variables) are available. Li et al. [4] integrated weather data with LSTM-like architectures to produce strong improvements for major crops. However, deep models are resource intensive and harder to interpret, limiting adoption for district-level APMC deployments. Hybrid models (e.g., ARIMA residuals combined with XGBoost) can harness both linear structure and non-linear residual modeling.

D. Gaps and Rationale

Key gaps include:

- Lack of reproducible, deployable pipelines that combine good data engineering with rigorous modeling for APMC data.
- Insufficient interpretability and policy-relevant outputs (e.g., uncertainty or feature-attribution) in many prior works.
- Few open benchmarks for vegetable forecasting at the district level.

This paper aims to bridge these gaps with a documented pipeline, emphasis on interpretability, and real-world deployment considerations.

IV. Dataset and Preprocessing

A. Data Source

We use daily modal price data from Agmarknet for Pune district markets spanning January 1, 2023 to June 30, 2025. The selected markets include: Pimpri (Pune), Moshi (Pune), Chakan (Khed), and Narayangaon (Junnar). Commodities considered: cabbage, cauliflower, and green chilli. Each record contains:

- Market identifier (string)
- Commodity & grade (string)
- ArrivalDate (YYYY-MM-DD)
- MinPrice, MaxPrice, ModalPrice (per quintal)

Raw datasets were downloaded as CSV dumps via Agmarknet APIs and consolidated into a unified schema.

B. Data Cleaning

APMC datasets present several issues: missing dates, duplicate entries, and occasional entry errors (e.g., zero or implausible prices). Our cleaning pipeline:

1) Remove duplicates by (Market, ArrivalDate, Commodity, Grade).
2) Parse dates and sort chronologically for each Market+Commodity+Grade group.
3) Impute missing modal prices: where possible, use forward-fill (previous day's modal price) to retain continuity; if missing at the beginning, use median series price for that market.
4) Winsorize modal prices at 95th percentile to reduce influence from anomalous spikes (but also log the winsorized events for possible later analysis).

C. Exploratory Data Analysis (EDA)

We examined descriptive statistics per commodity and market: mean, median, standard deviation, interquartile ranges, and autocorrelation functions (ACF). Typical observations:

- Strong weekly autocorrelation (7-day seasonality) for vegetables.
- Noticeable monthly and festival-season peaks.
- Heteroskedasticity: variance increases during supply shocks.

V. Feature Engineering

Effective forecasting hinges on appropriate feature construction. We create features grouped into: temporal, lagged & rolling, seasonal (Fourier), and market-contextual.

A. Temporal Features

- Year, Month, Day-of-Month, Day-of-Week (0–6)
- IsWeekend (boolean)
- Public-holiday indicator (per market, where available)

B. Lagged & Rolling Features

Lag features (calculated per market+commodity+grade group):

$$L_k(t) = \text{ModalPrice}(t - k), \quad k \in \{1, 2, 3, 7, 14\}.$$

Rolling features (7-day and 14-day windows):

$$\bar{Y}_w(t) = \frac{1}{w}\sum_{i=1}^{w} Y(t-i), \quad \sigma_w(t) = \text{stddev}(Y_{t-w+1:t}),$$

Exponential moving average (EMA) with span 7.

C. Fourier-Based Seasonal Encoding

To represent cyclicality without ordinal artifacts:

$$\text{Month\_sin}_k = \sin\left(\frac{2\pi k \cdot \text{Month}}{12}\right), \quad \text{Month\_cos}_k = \cos\left(\frac{2\pi k \cdot \text{Month}}{12}\right),$$

for $k = 1, 2$ (two harmonics).

D. Market Context Features

- PriceRange = MaxPrice - MinPrice
- MarketMedianPrice (rolling median across last 30 days across markets) as proxy for regional trends
- Grade indicators (FAQ, Local)

E. Feature Selection

Features with high multicollinearity were pruned. We used mutual information and preliminary XGBoost gain scores on a small validation fold to rank and select top 30–40 features.

VI. Mathematical Formulation of XGBoost

This section sets out formal derivations and equations used in the training algorithm.

A. Model Representation

We model the target as:

$$\hat{y}_i = \sum_{m=1}^{M} f_m(x_i), \quad f_m \in \mathcal{F},$$

where $\mathcal{F}$ is the space of regression trees.

B. Objective

At iteration t, the objective to minimize:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t).$$

Using second-order Taylor expansion of the loss around $\hat{y}_i^{(t-1)}$:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n}\left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \Omega(f_t),$$

with $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$.

C. Regularization Term

For a tree f_t with T leaves and leaf weight vector w, define:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1} w_j^2.$$

D. Optimal Weight per Leaf

For leaf j, let I_j be the set of instances assigned to the leaf. The optimal weight:

$$w_j^* = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda}.$$

The corresponding contribution to the loss:

$$\mathcal{L}_j = -\frac{1}{2}\frac{\left(\sum_{i\in I_j} g_i\right)^2}{\sum_{i\in I_j} h_i + \lambda} + \gamma.$$

E. Split Gain

For a candidate split of node I into left I_L and right I_R, the gain:

$$\text{Gain} = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{\left(\sum_{i\in I} g_i\right)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma.$$

A split is accepted if Gain > 0 (greater than minimum split gain threshold).

F. Learning Rate and Shrinkage

To reduce overfitting, predictions at each step are shrunk by learning rate η:

$$\hat{y}^{(t)} = \hat{y}^{(t-1)} + \eta f_t(x).$$

This slows down learning and cooperates with regularization to control model complexity.

VII. Methodology

A. Experimental Protocol

We split data chronologically: training set contains data until December 31, 2024; test set contains data from January 1, 2025 onward. Validation folds were generated via rolling-origin evaluation to tune hyperparameters.

B. Model Training

XGBoost hyperparameter grid:

- n_estimators: [200, 500, 800]
- learning_rate: [0.01, 0.05, 0.1]
- max_depth: [4, 6, 8]
- subsample: [0.6, 0.8, 1.0]
- colsample_bytree: [0.6, 0.8, 1.0]
- lambda (L2 reg): [1, 5, 10]

Grid search with time-series cross-validation on training data was used to pick optimal parameters. Final model used n_estimators=500, learning_rate=0.05, max_depth=6, subsample=0.8, colsample_bytree=0.8, lambda=5.

C. Baselines

We compared with:

- ARIMA: fitted individually per commodity series with AIC-based order selection.
- LSTM: single-layer LSTM with 64 hidden units, trained for 50 epochs, sequence window 14 days, trained per commodity.

D. Evaluation Metrics

Primary metrics: MAE and RMSE. In addition, we report MAPE (mean absolute percentage error) and quantify coverage of short-term directional accuracy (percentage of days where predicted change sign matches actual).

E. Algorithm Pseudocode

Algorithm 1 XGBoost Forecasting Pipeline

```
Input: Raw Agmarknet CSVs for markets and commodities
Data cleaning: deduplicate, parse dates, winsorize outliers
Feature engineering: temporal, lagged, rolling, Fourier, market-context
Chronological splitting: train (¡= 2024-12-31), test (¿= 2025-01-01)
Hyperparameter tuning via rolling-window CV on training data
Train final XGBoost model with chosen hyperparameters
Compute metrics (MAE, RMSE, MAPE) on test set
Serialize model artifacts and feature metadata
Deploy via Flask REST API for real-time inference
Monitor predictions and retrain monthly if drift detected
```

VIII. System Architecture and Deployment

We designed a modular architecture for data ingestion, preprocessing, model training, evaluation, and deployment. The compact horizontal flowchart below fits a full-width figure environment.

A. API Specification

The Flask endpoint accepts the following JSON payload


```
{
  "date":       "YYYY-MM-DD",
  "market":     "Pimpri",
  "commodity": "Cauliflower",
  "grade":      "FAQ"
}
```


and returns


```
{
  "predicted_modal": 1234.5,
  "model_version":    "v1.0",
  "confidence":       0.85
}
```


Confidence is derived from historical residual distributions (quantile-based).

IX. Experimental Results

A. Training Setup

All modeling conducted on a standard CPU instance (Intel Xeon, 16 cores, 64GB RAM). LSTM training used the same hardware (no GPU) to keep comparisons fair.

B. Quantitative Results

Table I summarizes detailed metrics on the test set (post-2025) for all three commodities.

XGBoost consistently achieves lower MAE and RMSE and better directional accuracy at much lower training time than LSTM (when both run on CPU).

C. Ablation Studies

We conducted ablation experiments to quantify the effect of feature groups:

- **No lags**: Remove all lagged features — MAE increases by 28% on average.
- **No rolling statistics**: Remove rolling mean/std — MAE increases by 12% on average.
- **No Fourier**: Remove Fourier encoding — MAE increases by 8%.
- **No winsorization**: Not capping outliers — RMSE increases by 18%.

This demonstrates the importance of temporal memory and robust outlier handling.

D. Interpretability

Feature importance (XGBoost gain) consistently ranked:

1) Lag1_Modal
2) RollingMean7
3) Month_cos (Fourier)
4) PriceRange
5) Market indicator (Pimpri)

We use SHAP on a subset of the test data to provide local explanations.

X. Case Study: Pune (Pimpri) Market

We present a focussed case study of Pimpri market for Cauliflower across Jan–Mar 2025. The model captured the increase in prices and predicted a short-term peak 3 days prior to the actual peak.

Key takeaways:

- Short-term (1–3 day) forecasts are most accurate (MAE ¡ 3% of mean price).

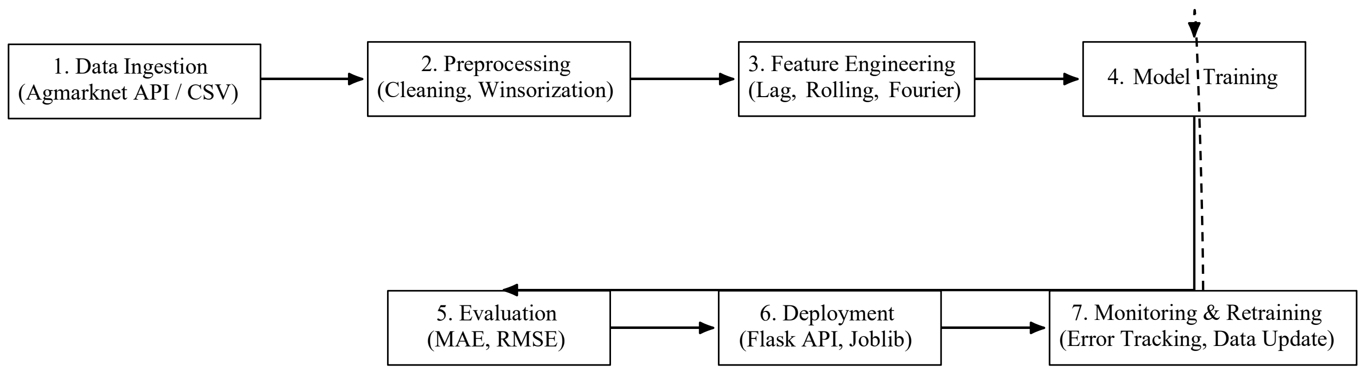

Fig. 1: Two-row system architecture illustrating the complete data pipeline from ingestion to retraining. The dashed arrow indicates periodic retraining based on monitoring feedback.

TABLE I: Test Set Performance (Post-2025) — XGBoost vs Baselines

Crop	Model	MAE ()	RMSE ()	MAPE (%)	Dir. Acc. (%)	Mean Price ()	N (days)
Cabbage	ARIMA	125.1	225.4	7.9	61.2	1150	7860
	LSTM	95.2	190.3	6.6	66.5	1150	7860
	XGBoost	**81.6**	**174.3**	**5.4**	**71.5**	1150	7860
Cauliflower	ARIMA	158.7	210.8	9.5	59.8	1660	7623
	LSTM	141.4	190.0	8.6	63.2	1660	7623
	XGBoost	**113.3**	**156.6**	**6.8**	**69.4**	1660	7623
Green Chilli	ARIMA	358.9	439.2	11.9	56.4	4260	6534
	LSTM	298.6	372.5	9.5	59.8	4260	6534
	XGBoost	**251.5**	**342.1**	**7.5**	**65.1**	4260	6534

- Multi-day horizon (7+ days) requires further exogenous inputs.

XI. Deployment Considerations and Operationalization

We built a lightweight Flask API serving serialized model artifacts. The inference pipeline:

1) Accepts market and date input.
2) Reconstructs features.
3) Applies scaling and feeds into the XGBoost model.
4) Returns predicted modal price with top contributing features.

For production, we recommend:

- Using a feature store (e.g., Feast).
- Containerizing API with Docker.
- Scheduling nightly retraining.

XII. Limitations

- Absence of exogenous variables restricts long-horizon forecasting.
- Dataset is limited to Pune district.
- The current model provides point forecasts.

XIII. Future Work

Future directions include:

- Integrating IMD weather feeds and fuel price series.
- Building hybrid ensembles.
- Extending to probabilistic forecasting.
- User studies with farmers and traders.

XIV. Policy Implications

Accurate short-term forecasts can inform government interventions, reduce post-harvest losses, and improve market transparency. Integration with e-NAM would enhance reach.

XV. Conclusion

This paper presents a comprehensive XGBoost-based forecasting pipeline for vegetable prices in Pune APMC markets. The model outperforms ARIMA and LSTM, achieving MAE of 81–251. The system is deployed with monitoring and retraining, making it suitable for real-world adoption.

Acknowledgment

The authors thank MIT Art, Design & Technology University for support. We acknowledge Agmarknet for providing open access to market data. We are grateful to Mr. Rohit Joshi (Assistant Professor) for feedback on deployment and optimization.

REFERENCES

[1] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications*, 4th ed., Springer, 2017.

[2] R. K. Paul, A. K. Mahto, *et al.*, "Machine learning techniques for forecasting agricultural prices: a case of brinjal in Odisha, India," *PLOS ONE*, vol. 17, no. 7, e0270553, 2022.

[3] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD*, 2016, pp. 785–794.

[4] S. Li, *et al.*, "Exogenous variable driven deep learning models for improved price forecasting of TOP crops in India," *Scientific Reports*, vol. 14, no. 1, 16803, 2024.