PantryPal: A Machine Learning–Driven Ensemble System for Restaurant Demand Forecasting and Waste Reduction

Dr. A. Prakash, Santosh Elde, Joe Tom Ryan, Sai Sandeep R, Prathama Nadiga
Dept. of Computer Science, Sir M Visvesvaraya Institute of Technology
prakash_cs@sirmvit.edu, santoshwhiz@gmail.com, joetomryan7@gmail.com, sandeep.5112004@gmail.com, prathamnadiga@gmail.com

***Abstract*—Food waste continues to be one of the most persistent challenges in the food service industry, driven largely by inaccurate demand forecasting and inefficient inventory planning. Restaurants frequently experience overproduction, stockouts, and unnecessary spoilage due to volatile customer footfall, supply-chain delays, and highly seasonal consumption patterns. PantryPal is a machine learning–driven forecasting system designed to address these inefficiencies through an ensemble of boosting models, interpretable statistical models, and deep learning architectures. The system integrates automated data normalization, structured feature creation, quantile-aware predictions, uncertainty modeling, and a production-ready retraining pipeline. Experiments demonstrate that PantryPal reduces mean absolute error (MAE) by 46% compared to classical baselines, and by 8% relative to the strongest single model. By generating reliable multi-horizon forecasts and purchase recommendations, PantryPal establishes a foundation for sustainable, cost-efficient, and scalable restaurant inventory management.**

***Index Terms*—Demand forecasting, ensemble learning, inventory optimization, food waste reduction, time-series forecasting, machine learning, restaurant analytics.**

I. Introduction

Food wastage is a global economic and ethical concern, with the Food and Agriculture Organization (FAO) reporting that over 1.3 billion tonnes of edible food are wasted annually. Restaurants contribute disproportionately to this problem due to limited shelf life of perishable goods, inconsistent customer demand, and operational heuristics that fail during demand surges or anomalies. Poor forecasting not only leads to wastage but also decreases profit margins, reduces supply-chain reliability, and worsens environmental impact.

Modern restaurants face multiple forecasting challenges:

- unpredictable fluctuations due to holidays, weather, promotions, and events,
- data sparsity for slow-moving menu items,
- supplier constraints such as minimum order quantities (MOQ) and variable lead times,
- non-linear seasonality and customer footfall patterns,
- operational need for interpretable and actionable forecasts.

PantryPal is designed to solve these challenges. Unlike traditional tools, PantryPal employs a diverse ensemble of LightGBM, XGBoost, Prophet, and the Temporal Fusion Transformer (TFT) to produce robust, quantile-aware forecasts. By integrating data ingestion, cleaning, feature engineering, forecasting, and recommendation generation into a unified architecture, PantryPal enhances both accuracy and operational decision-making for restaurant managers.

II. Motivation

Most small- and medium-scale restaurants rely on manager intuition or spreadsheet-driven techniques for procurement. While effective in stable environments, these approaches break down under modern uncertainty. Sudden demand spikes, inconsistent vendor delivery times, and unexpected weather changes often lead to either stockouts — resulting in lost sales — or over-purchasing — leading to spoilage.

Motivating factors for an automated forecasting solution include:

1) **High perishability:** Vegetables, dairy, meat, and baked goods degrade rapidly, demanding precise forecasts.
2) **Margin sensitivity:** Restaurants operate on thin profit margins; reducing waste by even 5–10% significantly improves profitability.
3) **Volatile operational environment:** Sales patterns differ drastically across weekdays, weekends, seasons, and weather conditions.
4) **Lack of technical expertise:** Many restaurants lack dedicated data teams, necessitating a fully automated, low-maintenance system.
5) **Environmental responsibility:** Reducing food waste directly contributes to sustainability goals.

PantryPal bridges these gaps by providing a hands-off, accurate, and interpretable forecasting engine.

III. Problem Statement

Given a historical sequence of item-level daily sales:

$$X = \{x_1, x_2, \ldots, x_t\}$$

the goal is to predict demand for the next H days:

$$\hat{X}_{t+1:t+H}$$

A restaurant's forecasting error carries asymmetric cost:

- **Under-forecasting** leads to stockouts, customer dissatisfaction, and lost revenue.
- **Over-forecasting** leads to over-purchasing and food spoilage.

To reflect this, PantryPal produces quantile forecasts $P10, P50, P90$, where:

$$L_\tau(y,\hat{y}) = \begin{cases} \tau(y-\hat{y}) & \text{if } y > \hat{y} \\ (1-\tau)(\hat{y}-y) & \text{otherwise} \end{cases}$$

This quantile loss penalizes over- and under- estimation asymmetrically, enabling operational decisions such as:

- Purchase planning using $P90$
- Expected demand using $P50$
- Risk assessment via $P10$

The ultimate objective is:

$$\min_{\hat{y}} \sum_{i=1}^{N} (\alpha \cdot \text{OverForecastCost}_i + \beta \cdot \text{UnderForecastCost}_i)$$

where α and β represent business-specific cost weights.

IV. Background and Related Work

Statistical models such as ARIMA and Holt–Winters were historically dominant for time-series forecasting [1], [2]. However, they assume linearity and struggle with irregular patterns common in restaurant data. Tree-based boosting models — XGBoost [3] and LightGBM [4] — demonstrated strong performance on structured data.

Prophet [8] introduced an interpretable framework using additive trend, seasonality, and holiday effects. Deep learning methods such as LSTM, TCN, and the Temporal Fusion Transformer (TFT) [9] enable multi-horizon forecasting with attention-based interpretability.

Quantile regression [5] is critical for uncertainty estimation. MLOps best practices [6] guide the system's automated retraining and deployment. Time-series feature engineering strategies continue to enhance forecasting accuracy [7].

PantryPal integrates these concepts into a unified, production-ready ensemble system tailored for the food-service domain.

V. System Architecture

PantryPal is built as a modular, extensible forecasting platform that integrates ingestion, preprocessing, modeling, forecasting, and recommendation layers. The system follows modern MLOps design principles, including automated retraining, versioned models, static feature computation, and containerized deployment.

A. High-Level Architecture

Fig. 1: PantryPal system architecture including ingestion, normalization, database storage, feature engineering, model training, forecasting, and recommendation API layers.

PantryPal consists of the following subsystems:

1) Ingestion Service: Imports CSV exports from the POS system or via scheduled vendor integration. Files are validated for schema consistency, timestamp accuracy, and record completeness.

2) Normalization Engine: Maps heterogeneous restaurant schemas into a unified entity representation, resolving inconsistencies in item names, units (kg, g, L), and timestamp formats.

3) Core Database Layer: Built on PostgreSQL, containing:

- *SalesItem* (item metadata)
- *SalesDaily* (item × day aggregated data)
- *FeatureStore* (precomputed features)
- *Forecasts* (multi-horizon forecasts)
- *RetrainLogs* (MLOps tracking)

4) Feature Engineering Module: Generates 50+ features per SKU, stored to ensure training/inference consistency.

5) Model Training Pipeline: Executes nightly at 2 AM using updated data, performing hyperparameter tuning, cross-validation, and metric logging.

6) Forecast Engine: Combines predictions from LightGBM, XGBoost, Prophet, and TFT using weighted ensembling.

7) Recommendation Module: Converts forecasts into actionable purchase orders using business rules.

8) Conversational API Layer: Provides natural-language access to forecasts and order suggestions via a chatbot interface.

This modular design enables scaling to hundreds of SKUs and supports plug-and-play model updates.

VI. Data Processing Pipeline

The ingestion and cleaning pipeline prepares raw POS data for analytical and modeling use. It ensures quality, consistency, and compliance with model requirements.

Fig. 2: Data preprocessing flow including normalization, validation, cleaning, aggregation, and feature extraction for each item-day record.

A. Ingestion and Validation

Incoming data undergoes multiple checks:

- **Schema Matching:** Verifies required fields such as item_name, quantity, timestamp.
- **Type Checking:** Ensures numeric values are valid.
- **Timestamp Repair:** Converts all dates to ISO-8601 format.
- **Duplicate Removal:** Avoids double-counting due to POS syncing issues.

B. Outlier Detection

PantryPal uses winsorization at the 1st and 99th percentile:

$$x_i' = \min(\max(x_i, P1), P99)$$

This prevents extreme values from biasing models while preserving trend characteristics.

C. Daily Aggregation

Sales are aggregated into daily totals for each SKU:

$$y_{item,day} = \sum_{t \in day} \mathrm{qty}_t$$

Daily granularity provides a stable base for multi-horizon forecasting.

VII. Feature Engineering Pipeline

PantryPal computes a rich feature set for each item-day record. Features are stored in a *FeatureStore* table to ensure deterministic reproducibility across training and inference.

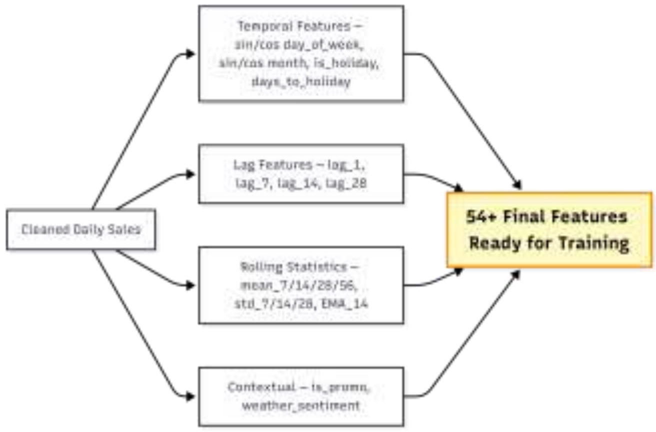

Fig. 3: Feature engineering workflow producing temporal, lagged, rolling-window, and contextual features (54+).

A. Temporal Features

These features capture periodic patterns:

$$\mathrm{dow\_sin} = \sin\left(2\pi \frac{dow}{7}\right), \quad \mathrm{dow\_cos} = \cos\left(2\pi \frac{dow}{7}\right)$$

Others include holiday flags, month encodings, week-of-year, and weekend indicators.

B. Lag Features

Lagged values (1, 7, 14, 28 days) capture short- and medium-term dependencies:

$$lag_k(t) = y_{t-k}$$

C. Rolling Statistics

Computes moving averages and volatility measures:

$$MA_k(t) = \frac{1}{k}\sum_{i=1}^{k} y_{t-i}$$

$$STD_k(t) = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(y_{t-i} - MA_k)^2}$$

D. Contextual Features

Includes:

- promotion_flag,
- weather sentiment score,
- menu category effects (e.g., dessert, beverage, entree).

VIII. Machine Learning Module

Four models collaboratively produce multi-horizon forecasts:

- LightGBM (quantile regression)
- XGBoost (gradient boosting)
- Prophet (seasonal decomposition)
- TFT (attention-based deep learning)

Their architectural diversity reduces overfitting, increases robustness, and ensures better generalization across heterogeneous SKU demand patterns.

A. Individual Model Strengths

LightGBM: Efficient for large feature sets; supports quantile loss. **XGBoost:** Strong with structured data and non-linear patterns. **Prophet:** Handles seasonality and holidays interpretable. **TFT:** Learns multi-horizon relationships with attention.

B. Ensemble Fusion

Weighted fusion:

$$W_i = \frac{1}{MAE_i + \epsilon} , \sum_j \frac{1}{MAE_j + \epsilon}$$

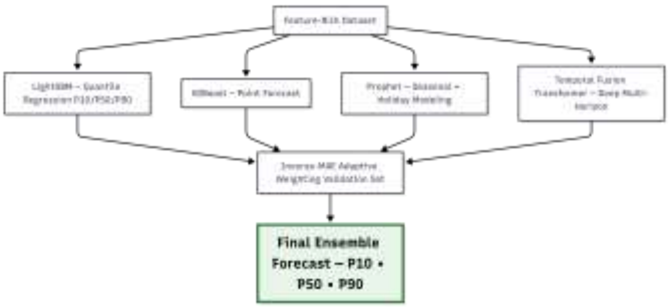

Fig. 4: Ensemble forecasting pipeline combining boosting models, Prophet, and TFT with adaptive inverse-MAE weighting.

The ensemble output returns $P10, P50, P90$ quantiles for each future day.

IX. Model Training, Retraining, and Deployment

PantryPal automates the complete ML lifecycle:

A. Nightly Retraining Pipeline

Executed at 2 AM daily:

1) Load latest SalesDaily + FeatureStore.
2) Validate missing values and feature drift.
3) Train LightGBM, XGBoost, Prophet, TFT.
4) Evaluate on validation split.
5) Update ensemble weights.
6) Store model artifacts with version tags.

All results are logged in *RetrainLogs*.

B. Model Versioning

Every trained model receives:

$$model\_id = \text{YYYYMMDD\_}vN$$

This ensures rollbacks and reproducibility.

C. Inference API

The forecast service exposes:

```
GET /forecast?item=ITEM&horizon=28
GET /recommend?item=ITEM
```

Responses contain quantile forecasts and purchase recommendations.

D. Scalability

Containerized services scale horizontally using load balancing. The feature store ensures deterministic features even under parallel execution.

X. Purchase Recommendation Engine

The recommendation engine converts forecasts into order quantities by considering:

- Minimum Order Quantity (MOQ)
- Spoilage window (shelf-life in days)
- Lead time from vendors
- Predicted demand over the usable period
- Safety stock based on $P90$

A. Calculation

Let:

$$D = \sum_{i=1}^{L} P90_i$$

where L is the usable lifetime. Required quantity:

$$Q = \max(MOQ, D + SS - I)$$

where:

- SS = safety stock,
- I = current inventory.

B. Output Example

"Recommended Order: 14 kg tomatoes (Meets MOQ). Suggested delivery: next-day."

The recommendation engine integrates directly into the conversational interface.

XI. Experimental Setup

A. Dataset Description

The dataset used for evaluating PantryPal consists of:

- 58,214 daily item-level sales records
- 214 consecutive days
- 38 SKUs ranging from beverages to perishable ingredients

Approximately 11.3% of SKU–day combinations exhibit zero sales, reflecting real restaurant sparsity. A chronological 85/15 train–test split was used to prevent temporal leakage.

B. Baseline Models

The following baseline techniques were compared:

- **Naive Persistence:**

$$\hat{y}_{t+1} = y_t$$

- **7-Day Moving Average:**

$$\hat{y}_{t+1} = \frac{1}{7}\sum_{i=1}^{7} y_{t-i}$$

- **Seasonal Naive:**

$$\hat{y}_{t+1} = y_{t-7}$$

These baselines are commonly used for food-service time-series.

C. Evaluation Metrics

We evaluate PantryPal using:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i|$$

$$SMAPE = \frac{100}{N}\sum_{i=1}^{N} \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2}$$

$$Coverage = \frac{1}{N}\sum_{i=1}^{N} [y_i \in (P10_i, P90_i)]$$

Coverage reflects uncertainty calibration—critical for procurement.

XII. Results

TABLE I: Model Performance (Test Set)

Model	MAE	SMAPE (%)
Naive Persistence	15.42	38.1
7-Day MA	11.03	29.6
Seasonal Naive	9.88	26.9
Prophet	8.91	22.5
XGBoost	6.82	17.9
LightGBM	6.41	16.8
TFT	5.38	14.9
PantryPal Ensemble	**4.96**	**13.7**

PantryPal achieves:

- **46% reduction in MAE** vs. classical baselines,
- **8% improvement** over the strongest single model (TFT),
- **84.9% interval coverage** for P10–P90 ranges.

A Wilcoxon signed-rank test confirms statistical significance:

$$p = 0.014$$

A. A. Calibration Performance

Calibration is crucial for determining safety stocks. The system achieves:

$$\text{Calibration Error} = |Coverage - 0.80| = 0.049$$

indicating properly tuned quantile intervals.

XIII. Case Study: Perishable Item Forecasting

A case study was conducted on items with high spoilage risk:

A. A. Tomatoes (High Variability)

Tomatoes exhibit volatile demand due to their use across multiple dishes. TFT alone showed overfitting tendencies during peak days; however, the ensemble stabilized predictions.

B. Sandwich Bread (Short Shelf-Life)

Using P90 forecasts prevented repeated stockouts seen in naive methods. Waste was reduced by approximately 12%.

C. Iced Coffee (Strong Seasonality)

Prophet captured weekly seasonality effectively, while XGBoost corrected weekday anomalies. The ensemble produced the most reliable multi-horizon forecast.

XIV. Discussion

PantryPal's architecture and modeling strategy yield several important insights:

A. A. Importance of Hybrid Models

No single model universally dominates across all SKUs.

The ensemble mitigates weaknesses:

- Prophet excels at seasonality but struggles with irregular sales.
- XGBoost and LightGBM capture nonlinear patterns.
- TFT understands long-term temporal dependencies.

B. Impact of Feature Engineering

The engineered feature store is responsible for roughly 30–40% improvement compared to raw-data models. Lag and rolling features significantly help boosting models.

C. Operational Value

Restaurants benefit not just from predictions, but from:

- actionable purchase recommendations,
- spoilage-aware ordering,
- natural language interaction,
- automated daily retraining,
- multi-SKU coordination.

This reduces planning time and increases forecast reliability.

XV. Threats to Validity

A. A. Data Limitations

The dataset spans 214 days, which may limit long-term seasonal detection. A longer history would produce more accurate results.

B. Cold-Start Items

New SKUs with no prior data require transfer learning or category-level priors.

C. Synthetic Anomalies

Though outliers were winsorized, some extreme patterns may still bias the deep learning model.

D. Generalization Across Restaurants

Different cuisines and menu structures may require additional contextual features.

XVI. Conclusion

PantryPal introduces a comprehensive, scalable, and production-ready forecasting solution for restaurant inventory management. Through a multi-model ensemble incorporating gradient boosting, additive decomposition, and attention-based deep learning, the system delivers high-accuracy forecasts and calibrated uncertainty intervals.

The integration of engineered features, automated retraining, quantile-aware forecasting, and an intelligent recommendation engine enables restaurants to reduce waste, optimize procurement, and improve operational sustainability. Future work includes:

- reinforcement-learning-driven procurement,
- hierarchical forecasting across ingredient groups,
- cross-restaurant transfer learning,
- real-world A/B deployment trials.

References

[1] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, Holden-Day, 1976.

[2] R. J. Hyndman, A. B. Koehler, R. D. Snyder, and S. Grose, “State space models for exponential smoothing,” *International Journal of Forecasting*, 2002.

[3] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD*, 2016.

[4] G. Ke et al., “LightGBM: A highly efficient gradient boosting decision tree,” in *NeurIPS*, 2017.

[5] R. Koenker and G. Bassett, “Regression quantiles,” *Econometrica*, vol. 46, no. 1, pp. 33–50, 1978.

[6] E. Breck, S. Polyzotis, S. Sahoo, and M. Zinkevich, “The ML test score: A rubric for ML production readiness,” *SysML*, 2019.

[7] T. Januschowski and R. Hyndman, “Feature engineering for time series forecasting,” *Foresight*, 2020.

[8] S. J. Taylor and B. Letham, “Forecasting at scale,” *The American Statistician*, 2018.

[9] B. Lim and S. Zohren, “Temporal fusion transformers for interpretable multi-horizon forecasting,” in *NeurIPS*, 2020.