SmartBite: AI-Driven Cafeteria Demand Forecasting and Menu Intelligence

1st Prashant Kumar
Department of Computer Science
Rishihood University
Sonipat, India
prashant.k23csai@nst.rishihood.edu.in

2nd Akula Jithendranath
Department of Computer Science
Rishihood University
Sonipat, India
akula.j23csai@nst.rishihood.edu.in

3rd Manasa Chinnam
Department of Computer Science
Rishihood University
Sonipat, India
manasa.c23csai@nst.rishihood.edu.in

***Abstract*—College cafeterias often rely on manual estimation for meal planning, leading to overproduction, food wastage, and increased operational costs. SmartBite addresses this challenge through an AI-driven system that accurately forecasts daily food demand at dish-level granularity using a complete machine learning pipeline. The project integrates a Tuned XGBoost model for per-dish forecasting, a Prophet model for aggregate trend analysis, and data-mining techniques such as Association Rule Mining and K-Means Clustering to uncover student preferences and demand patterns. Using a synthetically generated four-year dataset (2022–2025), SmartBite demonstrates the capability to model seasonality, event-driven fluctuations, and behavioral trends. The final system outputs actionable insights, menu suggestions, and seven-day forecasts, serving as a foundation for a real-time cafeteria management dashboard aimed at minimizing waste and improving resource planning.**

***Index Terms*—Food Demand Forecasting, Time-Series Analysis, XGBoost, Prophet, Association Rule Mining, Clustering, Synthetic Dataset, Cafeteria Optimization**

I. Introduction

College cafeterias routinely struggle with planning daily meal quantities due to reliance on manual estimation, experience-based judgment, or fixed weekly schedules. Such approaches fail to account for real-world fluctuations in student footfall, academic calendars, seasonal patterns, and special events. As a result, cafeterias face two persistent challenges: (1) *food overproduction*, leading to wastage and unnecessary operational cost, and (2) *underproduction*, resulting in shortages and reduced student satisfaction. Accurate demand forecasting is therefore essential for efficient resource planning and sustainable food operations.

The SmartBite system addresses this need by leveraging modern Artificial Intelligence (AI) and Machine Learning (ML) techniques to forecast cafeteria food demand at a granular, dish-specific level. The goal is to model how demand varies across meal types (breakfast, lunch, snacks, dinner), weekdays, seasonal cycles, and event-driven disruptions such as festivals, examinations, and campus activities. In addition to forecasting, SmartBite extracts patterns in student choices and recommends menu adjustments that align with observed preferences.

To support end-to-end experimentation, a synthetic dataset representing four academic years (2022–2025) was generated using a rule-based cafeteria simulator. The dataset incorporates realistic temporal patterns including weekly seasonality, holiday drops, event spikes, student count variation, menu rotations, and lag-based correlations in dish demand. This enables experimentation in scenarios where real cafeteria billing logs are unavailable.

The SmartBite pipeline consists of twelve structured modules ranging from data preparation to advanced forecasting and recommendation. The core technical contributions include: (1) a Tuned XGBoost model for per-dish time-series forecasting, (2) a Prophet model for aggregate daily demand analysis, (3) Association Rule Mining to discover frequently co-selected dishes, and (4) K-Means Clustering to categorize days into demand profiles such as normal, event-heavy, and vacation periods.

The outcomes of SmartBite provide actionable intelligence for cafeteria managers. Forecasts help reduce food wastage and prevent shortages, while preference patterns and cluster-driven insights guide menu customization and operational planning. The system's design forms the analytical backbone for a future interactive dashboard capable of real-time monitoring, forecasting, and decision support.

II. Literature Review

Recent research has demonstrated that data-driven approaches can significantly improve food demand forecasting and menu planning in institutional settings such as cafeterias and canteens. Traditional heuristic methods are increasingly being replaced by supervised machine learning, probabilistic forecasting, and pattern-mining techniques that explicitly model temporal structure and customer behaviour.

In [1], Reddy and Rajan evaluated several classical machine learning algorithms, including Random Forest and XGBoost, for daily meal demand forecasting in canteen environments. Their results showed that ensemble tree-based models consistently outperformed linear baselines, particularly when fed with carefully engineered calendar and event features. This motivates the use of gradient-boosted trees as a strong baseline for dish-level forecasting in SmartBite.

Prophet, originally proposed by Meta, has been widely adopted for business time-series applications because of its ability to decompose demand into trend, seasonality, and holiday effects with minimal manual tuning. Patel and Jain [2] applied Prophet to meal-demand data and demonstrated that explicit modelling of holiday calendars and event periods could capture abrupt spikes and drops in consumption more effectively than purely autoregressive models. This prior work motivates our use of Prophet as a complementary, aggregate forecaster for total daily servings in SmartBite.

Deep learning approaches have also been investigated for food industry forecasting. Nguyen et al. [3] explored Long Short-Term Memory (LSTM) architectures for time-series prediction in food production and logistics. Their study showed that LSTMs can capture complex nonlinear temporal dependencies when large amounts of historical data are available, but they typically require more computation, careful regularization, and longer development time than tree-based models. Given the current scope and the synthetic nature of our dataset, SmartBite focuses on interpretable, data-efficient ensemble models while leaving LSTM-based extensions as future work.

Beyond forecasting, understanding joint purchase behaviour is crucial for menu optimization. Sharma and Mehta [4] used association rule mining to discover frequently co-ordered food items in canteen systems, demonstrating that high-lift rules can guide the design of combo offers and targeted promotions. Their work supports our choice of Apriori and association rules to generate interpretable recommendations such as "students who select dish A often also take dish B" for each meal period.

Compared to these prior efforts, SmartBite provides an integrated pipeline that combines (i) dish-level forecasting with tuned XGBoost, (ii) aggregate daily forecasting with Prophet, (iii) preference mining via association rules, and (iv) day-level clustering for strategic planning. The project also introduces a configurable data-generation framework that simulates realistic cafeteria behaviour over multiple academic years, enabling experimentation in settings where real point-of-sale logs are not yet accessible. This integrated, end-to-end design positions SmartBite as both a practical decision-support tool and a reproducible reference architecture for AI-driven cafeteria management.

III. DATASET

A. Target Data Specification

The SmartBite system is designed around a structured, day-level cafeteria log in which each record corresponds to a specific *dish* served in a particular *meal* on a given calendar date. The core fields used in this project are:

- **Date**: Calendar date (2022–2025), stored as a proper `datetime`.
- **Day**: Day of week (Monday–Sunday).
- **Meal_Type**: Categorical meal slot: {*Breakfast, Lunch, Snacks, Dinner*}.
- **Dish_Name**: Name of the dish (144 unique vegetarian dishes).
- **Servings**: Number of portions of that dish served in the given meal on that date.
- **Student_Count**: Total number of students present on campus that day.
- **Event_Flag**: Binary indicator of a special college event (festivals, cultural days, fests, etc.; 1 = event day).
- **Vacation_Flag**: Binary indicator of vacation or holiday periods (long breaks, semester gaps; 1 = vacation day).
- **Total_Eaters**: Approximate total number of students who ate in the cafeteria that day (all meals combined).

This schema represents the *target* structure we would expect from real point-of-sale (POS) logs. The synthetic generator reproduces the same layout so that the models and pipeline can be used unchanged when real cafeteria data becomes available.

B. Target Data Specification

The SmartBite dataset is structured at the dish–meal–day level, where each record represents the number of servings for a specific dish during a specific meal on a particular date. Table I summarizes all fields used in modeling, feature engineering, preference mining, and forecasting tasks.

TABLE I
SMARTBITE RAW DATASET SCHEMA

Column Name	Description	Data Type	Example Value
Date	Calendar date in YYYY-MM-DD format.	Date	2024-03-15
Day	Day of the week (full name).	String	Monday
Meal_Type	One of four meal sessions: Breakfast, Lunch, Snacks, Dinner.	String	Lunch
Dish_Name	Name of the dish served in that meal session (100% vegetarian).	String	Paneer Butter Masala
Servings	Total number of servings taken for the dish (target variable; students may take multiple).	Integer	2700
Student_Count	Total students available on campus that day (base for calculations).	Integer	1120
Event_Flag	1 if the day has a campus event/festival/sports day, else 0.	Integer	0
Vacation_Flag	1 if the day is a vacation/holiday/semester break, else 0.	Integer	0
Total_Eaters	Number of students who actually participated in the meal.	Integer	890

C. Synthetic Dataset Generation Logic

Because historical POS data were not available at the time of development, we created a configurable simulator that generates realistic cafeteria demand over four academic years (2022–2025). The main design goals were: (i) preserve realistic magnitudes and variability for each meal, (ii) encode clear weekly and seasonal patterns, and (iii) allow explicit control of event and vacation effects so that forecasting models can be properly evaluated.

1) Base Structure and Dish Popularity: For each day in the range 1 January 2022 to 31 December 2025, the generator creates four meal slots (Breakfast, Lunch, Snacks, Dinner). For each meal, a rotating subset of the 144 available dishes is scheduled so that the daily menu is diverse but popular dishes recur more frequently.

Each dish is assigned a *base popularity score* (drawn from a skewed distribution), which determines its typical serving level. Base demand for a dish in a given meal is sampled from a log-normal–like distribution proportional to this score, with additional dish-specific noise to ensure that servings are non-negative integers and exhibit realistic dispersion.

2) Weekly Patterns, Events, and Vacations: To capture calendar effects, several multiplicative factors are applied:

- **Weekly pattern:** Day-of-week multipliers model recurring trends such as higher demand on weekdays and lower demand on weekends. In particular, Mondays and Fridays receive slightly higher factors, while Sundays are reduced.
- **Event days:** A synthetic event calendar is injected each year (festivals, college fests, celebrations). On these days, both `Event_Flag` and per-dish demand are boosted using event-specific multipliers, resulting in visible spikes in total servings.
- **Vacation periods:** Multi-day vacation blocks (semester breaks, long holidays) are created where `Vacation_Flag` is set to 1. During these periods, the number of students on campus and the servings for all dishes are sharply reduced.
- **Exam periods and mild seasonality:** Certain exam windows and seasonal effects (e.g., slightly lighter demand in peak summer) are simulated via smaller adjustments to the base multipliers.

Finally, a modest amount of random noise is added at the dish level so that even on similar calendar days, demand is not perfectly deterministic. This makes the prediction task closer to a real-world scenario.

3) Student Count and Cafeteria Participation: Daily `Student_Count` values are sampled around campus capacity using smooth trends plus noise, producing realistic ranges across the four academic years. The generator then computes `Total_Eaters` by multiplying `Student_Count` with a participation rate that depends on whether the day is normal, event-heavy, or vacation-like. Small random perturbations are added so that participation varies from day to day.

In theory, `Total_Eaters` should satisfy `Total_Eaters` $\leq$ `Student_Count` for every day. In practice, because of the added noise, a small number of synthetic rows violate this constraint. This inconsistency was detected during Cell 3 anomaly checks and is explicitly acknowledged; importantly, `Total_Eaters` is *not* used as an input feature in the forecasting models, so it does not introduce data leakage.

D. Final Dataset Statistics

After generation and loading in Cell 3 of the notebook, the final dataset used for analysis has the following characteristics:

- **Total records:** 33,567 dish-level rows covering four full years (2022–2025).
- **Temporal coverage:** 1 January 2022 to 31 December 2025, with every calendar day represented.
- **Categorical diversity:** 144 distinct dishes and 4 meal types (Breakfast, Lunch, Snacks, Dinner), with all 7 days of the week present.
- **Data quality:** No missing values and no duplicate rows were found. Logical checks confirmed that `Event_Flag` and `Vacation_Flag` are never simultaneously active, and that all servings are strictly positive.
- **Anomaly note:** 1,131 rows were identified where `Total_Eaters` exceeded `Student_Count` due to generator noise. Since `Total_Eaters` was treated as a descriptive variable and excluded from the feature set, these anomalies do not affect the forecasting or preference-mining models.

This synthetic dataset therefore provides a clean, well-structured testbed for evaluating AI-driven cafeteria demand forecasting, while remaining compatible with future integration of real POS data that follow the same schema.

IV. Methodology

This section describes the end-to-end SmartBite pipeline, the tools used, and the core data preparation and analysis steps that support the forecasting and preference-mining components.

A. Overall Pipeline

The SmartBite system is implemented as a modular pipeline, where each stage produces reusable artefacts (data files, models, and reports) for later stages. At a high level, the workflow is:

- **Synthetic Data Generation** (external script): creates a multi-year cafeteria log with realistic demand patterns, events, and vacations.
- **Data Loading & Quality Checks**: robust CSV loading, type enforcement, and anomaly checks.
- **Feature Engineering**: construction of calendar features, lagged/rolling demand features, and categorical encodings.
- **Modeling**: baseline and tuned tree-based regression models for per-dish demand prediction.
- **Preference Analysis**: association rule mining and clustering to discover menu combos and demand profiles.
- **Advanced Forecasting**: daily aggregate forecasting using Prophet.
- **Packaging**: export of all models, figures, and reports into a single project archive.

B. Environment and Tools

The implementation was carried out in **Google Colab** with **Google Drive** used for persistent storage of datasets, models, and reports. An initial setup cell creates a structured project directory (data, models, figures, reports, logs) under `/content/SmartBite` and mounts Drive when persistence is required.

The following Python libraries form the core technology stack:

- **Data & Utilities**: `pandas`, `numpy`, `pyarrow`, `joblib`.
- **Modeling**: `scikit-learn` (Linear Regression, Random Forest, Gradient Boosting), `xgboost` (XGBRegressor).
- **Advanced Forecasting**: `prophet` (for daily aggregate forecasting with holidays).

- **Preference Mining**: `mlxtend` (Apriori and association rules).
- **Explainability**: `shap` (SHAP value computation and plots).
- **Visualization**: `matplotlib` and `seaborn` (consistent styling and figure export).

Global random seeds are fixed (`SEED = 42`) across `random`, `numpy`, and the environment to ensure reproducible results and model comparisons.

C. Data Pre-processing and Feature Engineering

The raw SmartBite dataset (Table I) is first loaded from CSV, cast into appropriate types, and persisted as a Parquet file (`data_01_loaded.parquet`) for efficient re-use. Initial checks confirm that there are no missing values or duplicate rows. Logical anomaly checks also verify that:

- `Event_Flag` and `Vacation_Flag` are never both 1 on the same row.
- All `Servings` values are strictly positive.

A small number of rows violate the ideal constraint `Total_Eaters` $\leq$ `Student_Count` due to randomness in the generator. Since `Total_Eaters` is excluded from the feature set, these anomalies do not affect the models.

From this clean base, Cell 4 constructs a set of time-aware and categorical features suitable for tree-based regression:

- **Temporal features**:
  - `Year`, `Month`, `Week` (ISO week number),
  - `DayOfYear` (1–365/366),
  - `Day_of_Week_Num` (0 = Monday, …, 6 = Sunday),
  - `Is_Weekend` (1 for Saturday/Sunday, 0 otherwise).
- **Lagged and rolling features** (computed per (`Dish_Name`, `Meal_Type`) group):
  - `Servings_Lag_1`: servings for the same dish in the same meal at its previous occurrence.
  - `Dish_Avg_3`: rolling mean of the last three historical servings (using only past data).

  These features are filled with 0 for early occurrences that lack sufficient history, avoiding look-ahead leakage.
- **Categorical encodings**:
  - `Meal_Type_Code`: label-encoded version of `Meal_Type`.
  - `Dish_Code`: label-encoded `Dish_Name`.
  - `Day_Code`: label-encoded `Day` of week.

  The three encoders are persisted using `joblib` for use during forecasting and future deployment.

The final per-row feature vector comprises 14 input features:

- `Student_Count`, `Event_Flag`, `Vacation_Flag`,
- `Year`, `Month`, `Week`, `DayOfYear`,
- `Day_of_Week_Num`, `Is_Weekend`,
- `Meal_Type_Code`, `Dish_Code`, `Day_Code`,
- `Servings_Lag_1`, `Dish_Avg_3`,

with the target variable `Servings`. These are stored together in `data_02_featured.parquet` for downstream modeling.

D. Exploratory Data Analysis

Exploratory Data Analysis (EDA) is performed on the engineered dataset to validate synthetic patterns and identify key demand drivers.

- **Event and Vacation impact:** Bar plots compare average per-dish servings on Normal vs Event days, and Normal vs Vacation days, confirming substantial demand spikes on events and steep drops during vacations (see Fig. 1).
- **Demand by meal type:** Average servings per dish were computed across Breakfast, Lunch, Snacks, and Dinner. Contrary to common cafeteria intuition, Breakfast shows the highest per–dish demand in the SmartBite dataset, followed by Lunch (Fig. 2). This occurs because breakfast offers fewer dishes overall, resulting in higher per–dish serving counts despite lower student turnout in the morning.
- **Top dishes:** Aggregating servings over the full horizon reveals the top-20 most popular dishes (e.g., Chana Masala, Sambar, Rajma, Pav Bhaji), which are strong candidates for frequent rotation and special focus (Fig. 3).
- **Correlation heatmap:** A feature correlation matrix highlights that `Servings_Lag_1` and `Dish_Avg_3` are highly correlated with the target, quantitatively confirming the importance of lagged demand signals (Fig. 4).
- **Key predictor relationships:** Scatter plots of `Servings` vs `Servings_Lag_1` and `Servings` vs `Student_Count` show clear positive relationships, supporting the choice of these variables as primary predictors (Fig. 5).

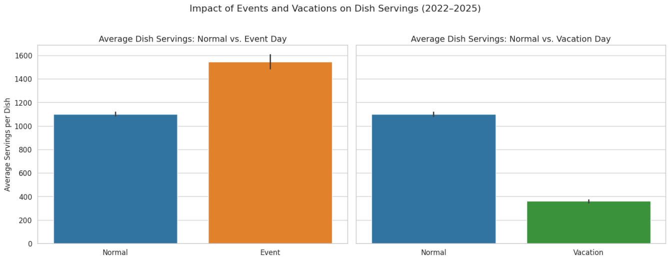

Fig. 1. Impact of events and vacations on average dish servings.

V. Forecasting Models

This section details the complete modeling workflow implemented in SmartBite, spanning baseline models, hyperparameter tuning, explainability, and final 7-day ahead forecasting. All experiments used strictly chronological splits to preserve temporal dependencies in cafeteria demand.

A. Experimental Setup

All forecasting experiments followed a unified evaluation protocol based on an **80/20 chronological split** of the featured dataset. The first 80% of dates (2022–2024 and early 2025) were used for training, while the remaining 20% of dates (2025) were reserved exclusively for testing.

The following evaluation metrics were used:

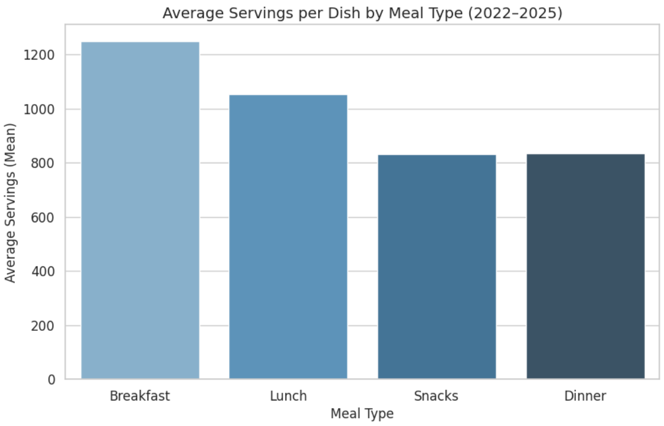

Fig. 2. Average servings per dish by meal type.

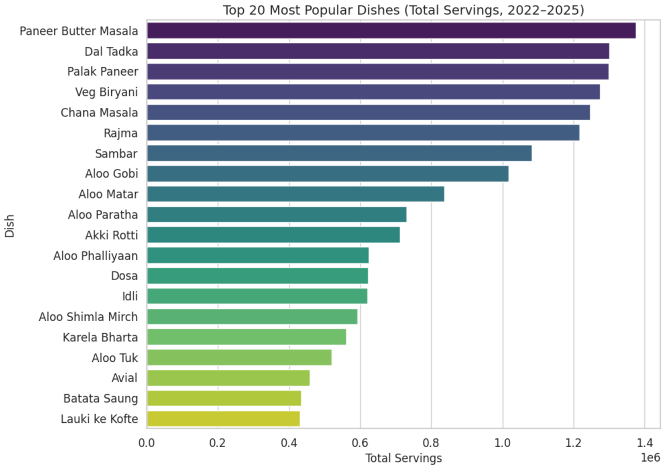

Fig. 3. Top-20 dishes by total servings across 2022–2025.

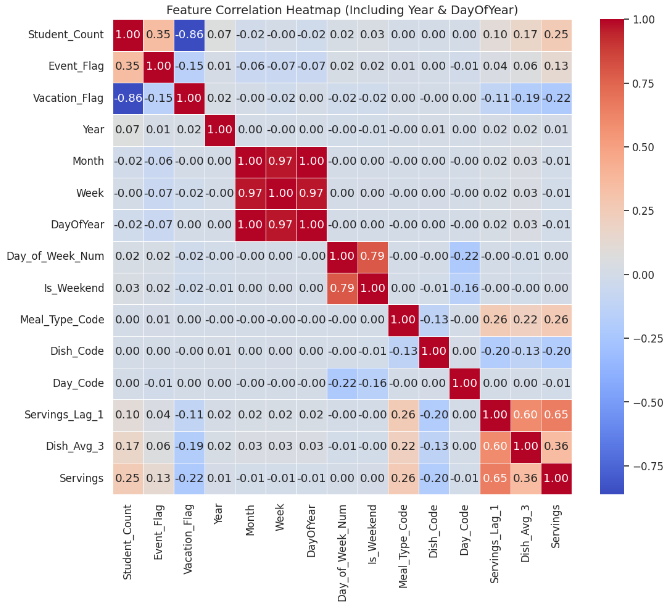

Fig. 4. Feature correlation heatmap including lagged and temporal features.

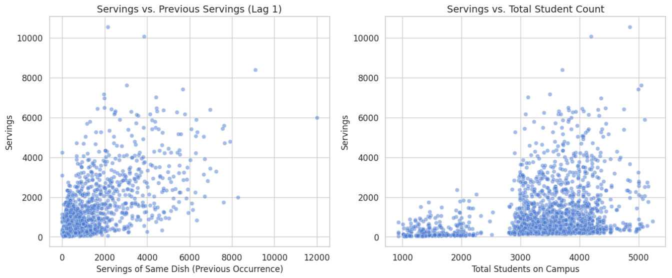

Fig. 5. Scatter plots showing relationships between servings, lagged demand, and student count.

- **MAE (Mean Absolute Error)**
- **RMSE (Root Mean Squared Error)**
- **MAPE (Mean Absolute Percentage Error)** – computed only for rows where the true value was > 1 to avoid division instability.

Negative predictions generated by certain regressors were clipped to zero to maintain physical validity (servings cannot be negative). This pipeline ensures fair comparison across all models.

B. Baseline Models

Four baseline regressors were trained in Cell 6 using the engineered features:

1) Linear Regression
2) Random Forest Regressor
3) Gradient Boosting Regressor
4) XGBoost Regressor

A consolidated comparison of baseline performance (RMSE) revealed:

- Random Forest achieved the best baseline performance.
- Gradient Boosting and XGBoost performed competitively.
- Linear Regression performed the worst due to its inability to model nonlinear seasonal patterns.

A summary table of baseline metrics was exported as `baseline_model_results.csv`, and the visual comparison figure is included in the report (placeholder: `baseline_comparison_figure.pdf`).

C. Hyperparameter Tuning

Hyperparameter tuning was conducted in Cell 7 using **TimeSeriesSplit** with $n\_splits = 3$ to preserve temporal ordering during cross-validation.

Two models were tuned:

- **Random Forest** (n_estimators, max_depth, min_samples_split, etc.)
- **XGBoost Regressor** (learning_rate, max_depth, subsample, colsample_bytree, n_estimators)

Randomized search over each hyperparameter space resulted in two optimized models. Final evaluation on the 20% test set demonstrated:

- Tuned XGBoost achieved the lowest RMSE and MAPE.
- Tuned Random Forest was competitive but slightly inferior.

Thus, the **tuned XGBoost model was selected as the final production model**. The final comparison figure (baseline vs tuned models) is referenced as:

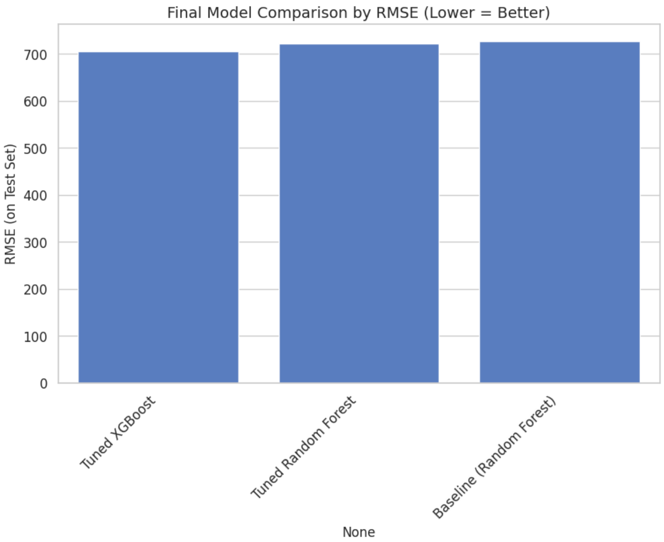

Fig. 6. Final RMSE comparison between baseline models, tuned Random Forest, and tuned XGBoost.

D. Model Explainability

Explainability analysis (Cells 7 and 7b) was conducted using:

- **XGBoost feature importance**
- **SHAP (SHapley Additive Explanations)**

The top contributing features included:

- **Servings_Lag_1** — strongest predictor of next-day demand.
- Dish_Code and Meal_Type_Code — capturing categorical dish identity and meal slot.
- Student_Count — capturing population shifts.
- Event_Flag and Vacation_Flag — capturing structural demand spikes and drops.

Three key explainability visuals are included:

These analyses validate the correctness of the engineered time-series features and confirm the model's reliance on temporal continuity.

E. Future Forecasting at Dish Level

Cell 8 generated a **7-day forecast** spanning **1 January 2026 to 7 January 2026**. To construct these predictions, the pipeline:

1) Re-loaded all label encoders used during training.
2) Generated all temporal features for the forecast horizon.
3) Merged the most recent historical Servings_Lag_1 and Dish_Avg_3 values for each dish–meal pair.

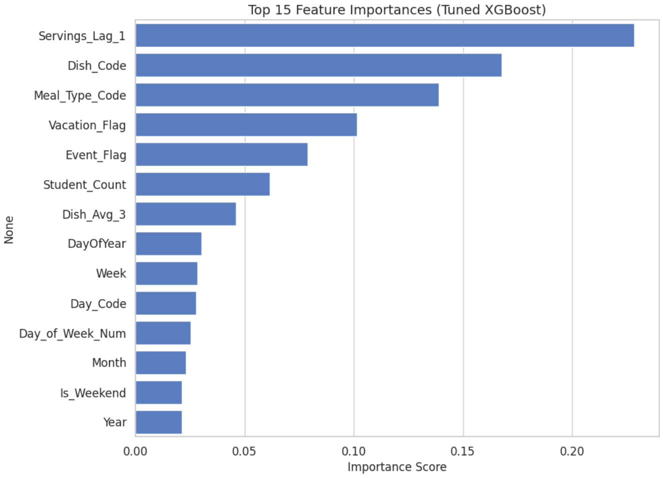

Fig. 7. Top feature importances from the tuned XGBoost model.

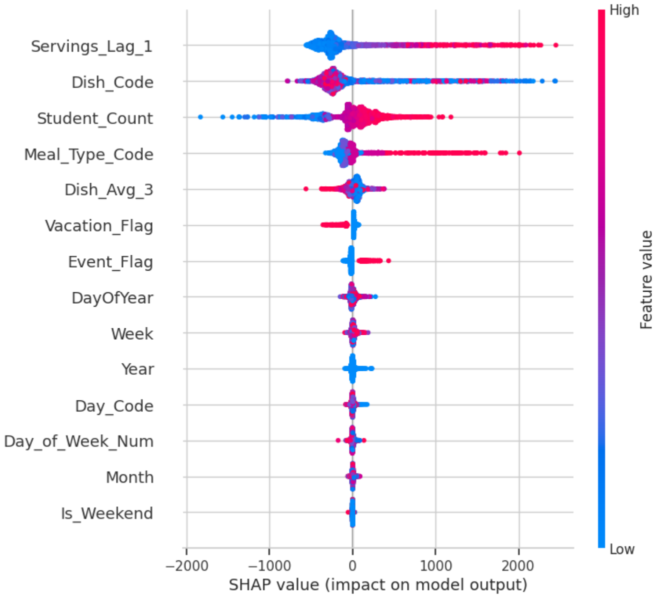

Fig. 8. SHAP summary plot for the tuned XGBoost model.

4) Assumed fixed Student_Count from the most recent observed day.
5) Set Event_Flag and Vacation_Flag to zero (default scenario).

The tuned XGBoost model predicted dish-level servings for all dish–meal combinations.

Key aggregated insights:

- **Snacks and Lunch** showed the highest mean predicted servings.
- Top predicted dishes included Chana Masala, Sambar, Rajma, Aloo Gobi, and Palak Paneer.
- A complete forecast dataset was exported to `forecast_next7days.csv` for dashboard integration.

A summary visualization of average servings per meal type

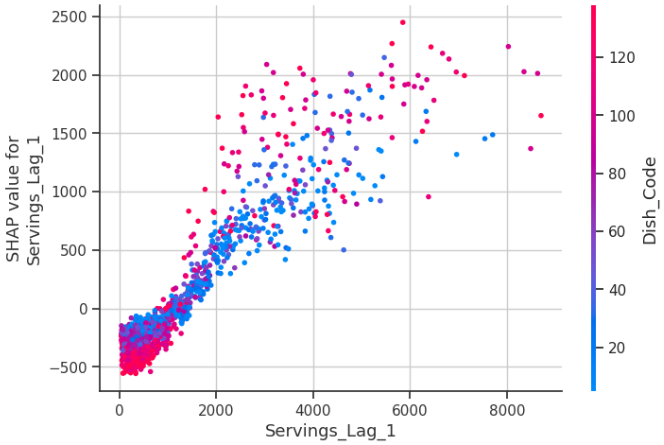

Fig. 9. SHAP dependence plot for the most important feature: Servings_Lag_1.

is included below:

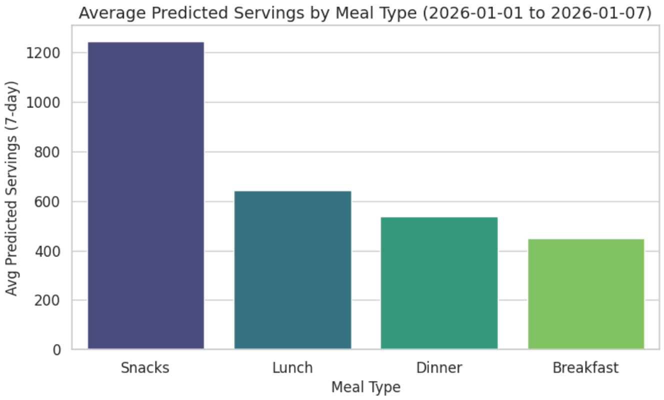

Fig. 10. Average predicted servings per meal type for the 7-day forecast window.

This section completes the per-dish forecasting component of SmartBite.

VI. Preference Analysis and Menu Intelligence

Beyond forecasting per-dish demand, SmartBite incorporates two analytical frameworks to understand student food habits and to assist cafeteria managers in optimizing menu design: (1) Association Rule Mining and (2) Daily Demand Clustering. These analyses reveal latent behavioral patterns that cannot be captured through forecasting alone.

A. Association Rule Mining

Association rules were generated in Cell 9 using the **Apriori** algorithm from the `mlxtend` library. For each meal type (Breakfast, Lunch, Snacks, Dinner), transactions were constructed as:

> **One transaction = All dishes consumed in a meal slot on a given day.**

This produced 1461 transactions per meal across four years (2022–2025). Apriori parameters were selected to balance sparsity and interpretability:

- **min_support** = 0.003 (appears in at least 0.3% of that meal's transactions)
- **min_confidence** = 0.40
- **min_lift** = 1.10

Frequent itemsets were computed separately for each meal type, and all resulting rules were combined into a unified rule base containing over **30,000** valid associations. Rules were ranked primarily by **lift**, which measures how much more frequently two dishes co-occur than expected by random chance.

An example visualization is shown in Fig. 11, illustrating support, confidence, lift, and meal category:

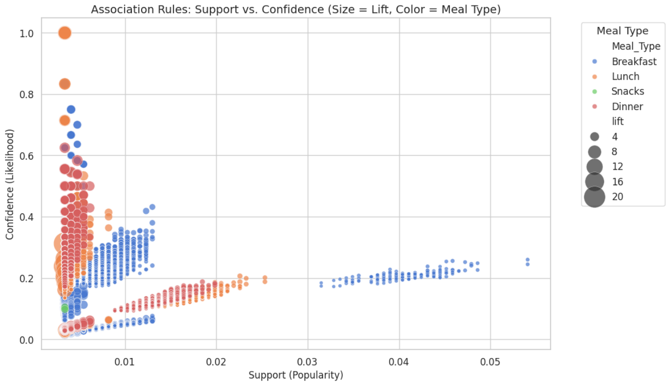

Fig. 11. Association rule scatter plot (Support vs Confidence), bubble size = Lift, color = Meal Type.

Examples of strong rules identified include:

- For Breakfast: *Antecedent*: {Akki Rotti, Pongal} *Consequent*: {Thepla} Confidence ≈ 0.83, Lift ≈ 4.44.
- For Lunch: *Antecedent*: {Mushroom Do Pyaza, Aloo Gobi} *Consequent*: {Kottu, Lauki ke Kofte} Lift $>$ 20.

Using these rules, a menu recommendation file `menu_recommendations_from_rules.csv` was produced containing textual, human-readable suggestions such as:

> *"[Lunch] When students take Aloo Gobi and Lauki ke Kofte, they also often take Mushroom Do Pyaza (lift = 15.38). Suggest pairing these dishes in the Lunch menu or offering them as a combo."*

These insights assist cafeteria planners in designing combo offers, optimizing food placement, and identifying dishes that naturally complement each other.

B. Daily Demand Clustering

To uncover macro-level behavioral patterns, Cell 10 performed clustering on daily aggregated consumption patterns.

Feature Construction: A pivot table was built where each row corresponds to a single day with the following numerical attributes:

- Total Breakfast Servings
- Total Lunch Servings
- Total Snacks Servings
- Total Dinner Servings
- Total Daily Servings (sum of all meals)
- Student_Count

These features were standardized using **StandardScaler**. K-Means was applied for $k = 2 \ldots 10$, and an elbow plot (Fig. 12) suggested **k = 4** as the optimal clustering value.

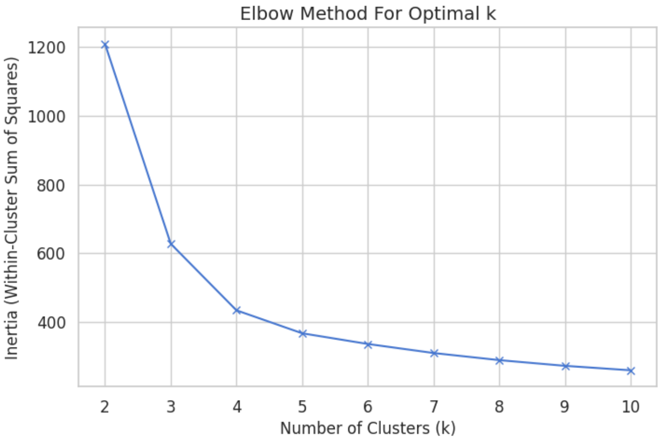

Fig. 12. Elbow method identifying $k = 4$ as the optimal number of clusters.

Cluster Profiles: The four clusters corresponded to meaningful operational categories:

- **Cluster 0: Normal-Low Demand Days** Moderate attendance, consistent meal patterns.
- **Cluster 1: Event-Heavy High-Demand Days** Very high student turnout and elevated consumption across all meals.
- **Cluster 2: Vacation-Like Low Days** Minimal attendance, sharp drops in all meal types.
- **Cluster 3: Normal-High Demand Days** Above-normal consumption driven by regular academic activity.

Normalized cluster profiles are shown in Fig. 13.

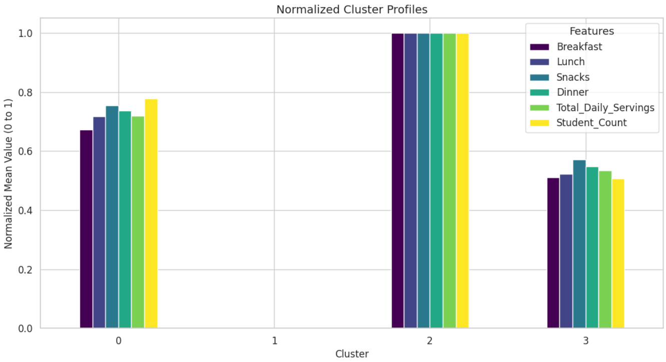

Fig. 13. Normalized feature profiles for each of the four daily demand clusters.

A PCA scatter plot (Fig. 14) illustrates the separation between clusters and validates the presence of distinct daily demand regimes.

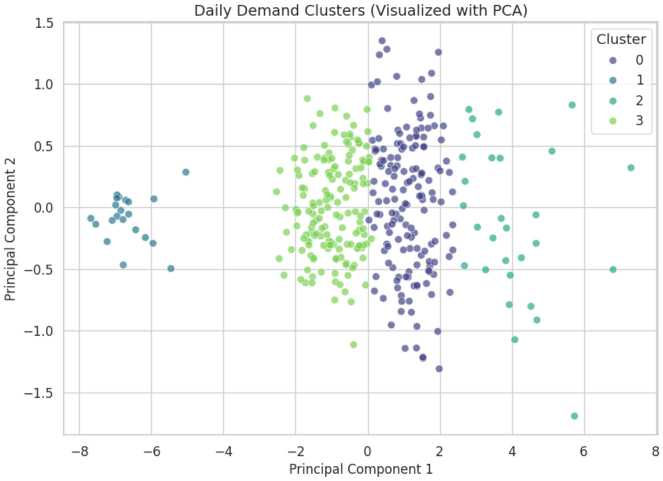

Fig. 14. PCA scatter plot showing the separation between the four demand clusters.

Strategic Recommendations: For each cluster, actionable recommendations were generated:

- **High-demand clusters**: increase batch sizes; prioritize core dishes; avoid under-preparation.
- **Event-heavy cluster**: include premium dishes or special items.
- **Vacation clusters**: reduce variety; minimize batch sizes to prevent waste.
- **Normal days**: follow standard production with small adjustments based on dominant meal type.

All recommendations were compiled in `cluster_strategy_recommendations.csv`.

VII. Advanced Top-Down Forecasting with Prophet

In addition to dish-level forecasting with XGBoost, SmartBite employs a top-down forecasting model using **Prophet** to predict overall daily demand. This provides a high-level view of cafeteria load, complementing the fine-grained per-dish forecasts.

A. Model Configuration

Cell 11 aggregates the transactional dataset to a daily level, computing `Total_Servings` as the sum of all dish servings for each date. This aggregate is used as the target variable:

- **Target:** Daily total servings $y(t)$.
- **Input:** Calendar date t plus holiday effects.

Two binary columns from the original dataset, `Event_Flag` and `Vacation_Flag`, are converted into a *holidays* dataframe in Prophet: each day with `Event_Flag = 1` is labeled as an *event* holiday, and each day with `Vacation_Flag = 1` is labeled as a *vacation* holiday.

The Prophet model is configured as:

- **Seasonality Mode:** Multiplicative (to allow seasonal effects to scale with demand).
- **Seasonalities:** Yearly and weekly enabled; daily seasonality disabled.
- **Holidays:** Custom holiday component encoding event and vacation days.

This configuration allows the model to capture long-term growth, weekly patterns, and sharp shifts due to events and vacations.

B. Train/Test Split and Evaluation

Consistent with the dish-level experiments, a chronological 80/20 split is applied at the *daily* level:

- **Training Period:** 2022-01-01 to 2025-03-13.
- **Testing Period:** 2025-03-14 to 2025-12-31.

Standard regression metrics are computed over the test horizon:

- **MAE** (Mean Absolute Error): 2579.93
- **RMSE** (Root Mean Squared Error): 3136.49
- **MAPE** (Mean Absolute Percentage Error): 13.06%

These values indicate that the model captures aggregate load with reasonably low percentage error over a long test horizon that spans multiple semesters, events, and vacation periods.

An overview of the forecast versus actual values, including the chronological train/test split, is shown in Fig. 15.

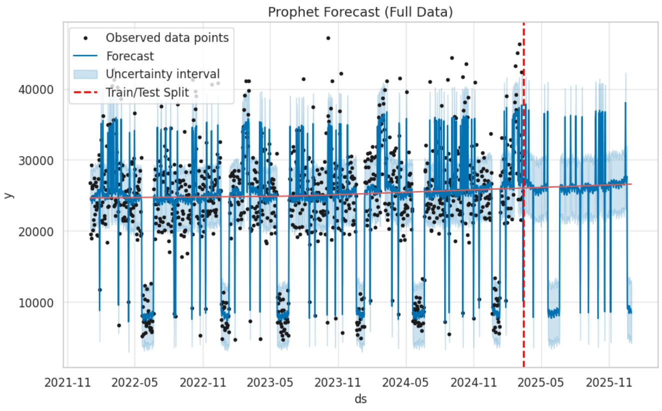

Fig. 15. Prophet forecast of total daily servings with train/test split marked.

C. Interpretation of Components

One of Prophet's main strengths is its interpretable decomposition into trend, seasonality, and holiday effects. The component plots (Fig. 16) summarize how SmartBite demand evolves over time.

Key observations include:

- **Trend:** A gradual upward trajectory from 2022 to 2025, reflecting increasing student strength and engagement over successive semesters.
- **Weekly Pattern:** Clear day-of-week effects, with certain days (e.g., mid-week and event-linked weekdays) exhibiting systematically higher total servings than others.
- **Holiday Effects:**
  - *Event days* show positive spikes, indicating substantially higher demand driven by festivals, cultural programs, or sports events.
  - *Vacation days* produce pronounced negative adjustments, capturing the sharp drop in demand during semester breaks and holidays.

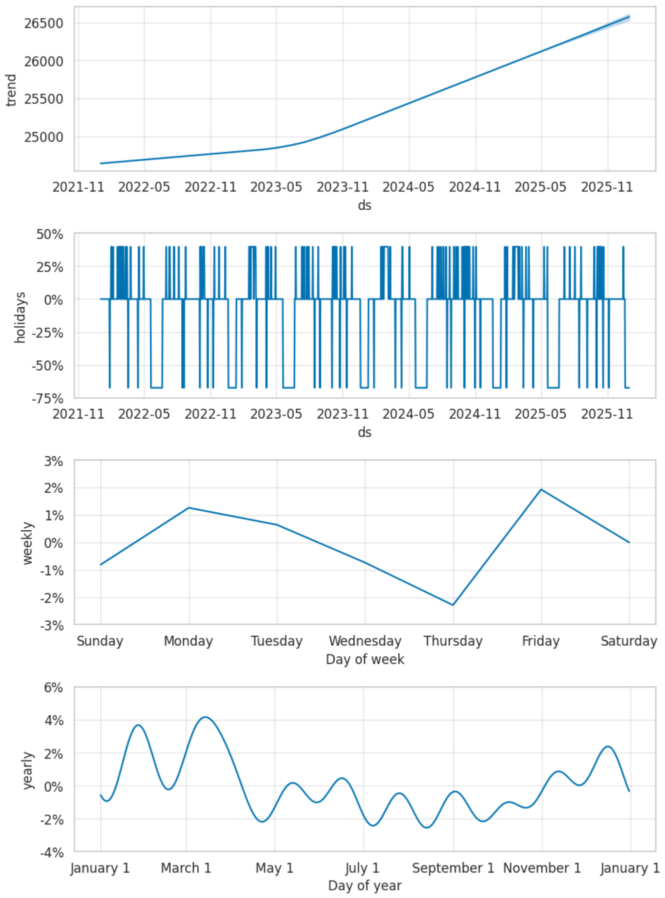

Fig. 16. Prophet components: long-term trend, weekly seasonality, and holiday effects (events and vacations).

These components validate the synthetic data design and confirm that event and vacation flags are meaningful drivers at the aggregate level.

D. Comparison to XGBoost

The Prophet model operates at a **top-down, aggregate** level, whereas the tuned XGBoost model focuses on **bottom-up, per-dish** forecasting. Their roles in SmartBite are complementary:

- **Prophet (Top-Down):**
  - Ideal for planning overall kitchen capacity, staffing, and high-level procurement.
  - Provides transparent explanations of trend, weekly seasonality, and holiday adjustments.
  - Slightly less precise for fine-grained menu allocation, since it forecasts total daily servings, not dish-level distributions.

- **XGBoost (Bottom-Up):**
  - Delivers higher accuracy at the dish level by exploiting lag features, dish codes, and student counts.
  - Supports operational decisions such as batch sizes per dish and menu adjustments per meal type.
  - Less inherently interpretable in terms of explicit seasonal components, but complemented by feature importance and SHAP analysis.

In practice, SmartBite uses Prophet to estimate the *total load* on the cafeteria and XGBoost to *allocate* that load across individual dishes and meal slots. Together, they form a coherent forecasting stack that is both accurate and explainable, supporting strategic planning as well as day-to-day operational decisions.

VIII. Discussion, Limitations and Future Work

A. Discussion

The SmartBite pipeline combines multiple analytical components to address the four objectives defined in Section **??**:

- **Demand Forecasting:** The tuned XGBoost model at the dish level (Section **??**) provides accurate short-term predictions of servings for each dish, meal type, and day. This directly supports the goal of forecasting daily food demand and reducing both shortages and wastage.
- **Traffic & Event-Based Demand:** Temporal features (Year, Month, Week, DayOfYear, weekday, weekend) and context flags (Event_Flag, Vacation_Flag) allow the models to learn demand shifts during exams, festivals, and semester breaks. The Prophet model further isolates these effects at the aggregate level, showing how events increase and vacations decrease total demand.
- **Student Preferences & Menu Adjustments:** Association rule mining (Section **??**) uncovers which dishes are commonly consumed together, providing a basis for combo offers and menu co-location. Daily demand clustering groups days into demand regimes (normal-low, normal-high, event-heavy, vacation-like), enabling menu strategies tailored to the expected demand type.
- **Actionable Insights:** All steps produce concrete artefacts (CSV reports, plots, and a packaged project zip) that can directly support an interactive dashboard. Together, they enable cafeteria managers to see *what* to cook, *how much* to prepare, and *how* to structure the menu for a given day type.

Using a synthetic dataset was particularly helpful in this first iteration: it allowed controlled experiments where the impact of events, vacations, and weekly patterns is known by design, making it easier to verify that the models recover the intended structure and to debug the pipeline end-to-end.

B. Limitations

Despite its strengths, the current SmartBite implementation has several limitations:

- **Synthetic Data Only:** All experiments are conducted on a synthetic dataset that emulates realistic cafeteria behaviour but has not yet been validated against real POS/billing logs. Model performance may change once exposed to noisy, real-world data.
- **Model Scope:** Although the proposal mentioned deep learning (e.g., LSTM), the implemented models are classical ML techniques (tree ensembles and Prophet). Temporal deep networks could capture more complex dependencies and long-range patterns.
- **Forecast Assumptions:** Future weekly forecasts at the dish level assume a fixed Student_Count (last observed) and no new events or vacations. Real-world deployment would require dynamic updates of student strength and accurate event calendars.
- **Simplified Behavioural Assumptions:** The synthetic generator assumes homogeneous student preferences within global popularity scores and cluster-level patterns. Real cafeterias may exhibit more nuanced behaviours (e.g., dietary constraints, price sensitivity, club-level events).

These limitations highlight that SmartBite is currently a *validated prototype* and not yet a production system.

C. Future Work

Several directions can further enhance SmartBite and bring it closer to deployment:

- **Integration with Real Data:** Replace or augment the synthetic dataset with real cafeteria transaction logs, retrain the models, and compare performance. This will also enable better calibration of event and vacation effects.
- **Advanced Time-Series Models:** Explore LSTM, Temporal CNN, and hybrid architectures as highlighted in the literature, using the same time-series split and feature set for fair comparison with XGBoost and Prophet.
- **Interactive Dashboard:** Implement the planned Streamlit dashboard so cafeteria managers can interactively explore forecasts, cluster profiles, menu rules, and what-if scenarios (e.g., adding new events or changing student strength).
- **Online Learning & Monitoring:** Introduce mechanisms for continuous retraining or incremental learning as new days are observed. Add monitoring for forecast error, drift in demand patterns, and automatic alerts when models require recalibration.
- **Richer Behavioural Signals:** Incorporate additional features such as price changes, feedback ratings, or special diet counters (e.g., Jain/vegan) to refine preference analysis and improve fairness and inclusivity in menu planning.

By following these directions, SmartBite can evolve from a controlled simulation-based study into a data-driven decision support system ready for deployment in real college cafeterias.

IX. Conclusion

The SmartBite project set out to address a critical operational challenge faced by university cafeterias: accurately predicting daily food demand to reduce wastage, prevent shortages, and streamline resource planning. By constructing a complete data-to-insight pipeline—including feature engineering, dish-level forecasting, association rule mining, clustering, and top-down Prophet analysis—the system demonstrates how machine learning can transform cafeteria operations into an evidence-driven process.

Across the 2022–2025 synthetic dataset, the tuned XGBoost model achieved strong predictive performance at the dish level, while the Prophet model uncovered seasonal and event-driven demand shifts at the aggregate level. The preference analysis components further revealed meaningful relationships between dishes, highlighting smart menu pairings and distinct daily demand regimes. Together, these insights provide powerful decision-support signals that directly enhance meal preparation, reduce operational waste, and improve student satisfaction.

SmartBite is not only a solution for a single cafeteria but a reusable framework. Its modular pipeline—data ingestion, time-series modeling, preference mining, and visual analytics—can be applied to other campuses, canteens, and institutional food services with minimal adaptation. With future extensions such as real POS data integration, deep-learning forecasting models, and deployment via an interactive dashboard, SmartBite can evolve into a scalable, production-ready platform for data-driven food service management.

References

[1] K. Reddy and S. Rajan, “Food Demand Forecasting using Machine Learning Techniques,” *International Journal of Computer Applications*, 2021.

[2] S. Patel and A. Jain, “Forecasting Meal Demand using Facebook Prophet,” in *Proc. IEEE Conference on Data Science and AI Applications*, 2022.

[3] T. Nguyen, M. Chen, and L. Hu, “Deep Learning for Time Series Forecasting in Food Industry,” *Applied Computing and Informatics*, 2023.

[4] R. Sharma and P. Mehta, “Menu Optimization using Association Rules in Canteen Systems,” *Journal of Emerging Technologies in AI*, 2021.