rishihood university X Newton School of Technology

SmartBite

AI-DRIVEN CAFETERIA DEMAND FORECASTING

Presented By:

Prashant Kumar (230101)

Akula Jithendranath (230120)

Manasa Chinnam (230078)

PROBLEM STATEMENT

College cafeterias prepare hundreds of meals every day, but planning is mostly guesswork. This leads to:

- Overproduction → Food wastage, higher costs.
- Underproduction → Shortages, long queues, student dissatisfaction.
- Demand varies heavily by:
  - Meal type (Breakfast vs Lunch vs Dinner)
  - Weekdays vs weekends
  - Events like festivals, sports days, fests
  - Vacations/exam breaks

Core Problem

Cafeterias do not have a reliable way to forecast daily food demand at the dish level.

- Need accurate forecasting + menu intelligence.

PROJECT OBJECTIVES

1. Predict Daily Food Demand (Dish-Level Forecasting)
   - Forecast servings required for each dish in Breakfast, Lunch, Snacks, and Dinner.
   - Use historical trends, lag features, dish popularity, and student count.

2. Model Event-Based and Seasonal Demand
   - Capture demand spikes during college events/fests.
   - Lower demand during vacations and exam breaks.
   - Incorporate seasonality: weekday patterns, monthly variations, yearly cycles.

3. Analyze Preferences & Suggest Menu Adjustments
   - Identify dishes commonly taken together (using Apriori rules).
   - Detect different types of days using clustering
(Normal-Low, Normal-High, Event-Heavy, Vacation-like).

4. Provide actionable insights for cafeteria management.

DATASET OVERVIEW

Dataset Structure:
Synthetic dataset created to simulate 4 years of cafeteria operations (2022–2025).
Each row represents: one dish served in one meal on one day.
Columns (9 total):
Total size: 33,567 rows × 9 columns.

```
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Date            33567 non-null  datetime64[ns]
 1   Day             33567 non-null  object
 2   Meal_Type       33567 non-null  object
 3   Dish_Name       33567 non-null  object
 4   Servings        33567 non-null  int64
 5   Student_Count   33567 non-null  int64
 6   Event_Flag      33567 non-null  int64
 7   Vacation_Flag   33567 non-null  int64
 8   Total_Eaters    33567 non-null  int64
dtypes: datetime64[ns](1), int64(5), object(3)
```

DATASET LOGIC

Synthetic Dataset Generation Logic:

To make the dataset realistic, multiple rules were applied:

- Base Popularity Scores:
- Each dish has a predefined popularity that drives its expected demand.
- Meal-Level Behavior:
  Breakfast usually has fewer students → lower base servings.
  Lunch and Dinner generally higher.
- Weekly Patterns:
  - Mondays/Fridays slightly higher demand
  - Weekends sometimes lower Students Count
- Event Days (≈10% days): Demand increased +40% across meals + special dish boosts.
- Vacation Periods: Demand reduced 60–80%; fewer dishes served.
- Random Noise: Added to avoid perfect linear patterns and mimic real-life variability.
- Logical Constraints:
  - Servings ≥ 0
  - Total_Eaters ≤ Student_Count (mostly enforced; a few anomalies ignored since feature unused)

FEATURE ENGINEERING

1. Temporal Feature Engineering

Added multiple date-based features to capture seasonality & weekly patterns:

- Year, Month, Week, DayOfYear
- Day_of_Week_Num (0–6)
- Is_Weekend (Sat/Sun)

These help the model learn periodic demand cycles.

2. Lag & Rolling Features (Most Important Predictors)

Created dish-level time-series features:

- Servings_Lag_1: Yesterday's servings for the same dish
- Dish_Avg_3: 3-day moving average of servings

These capture demand momentum, contributing the most to XGBoost performance.

3. Categorical Encoding

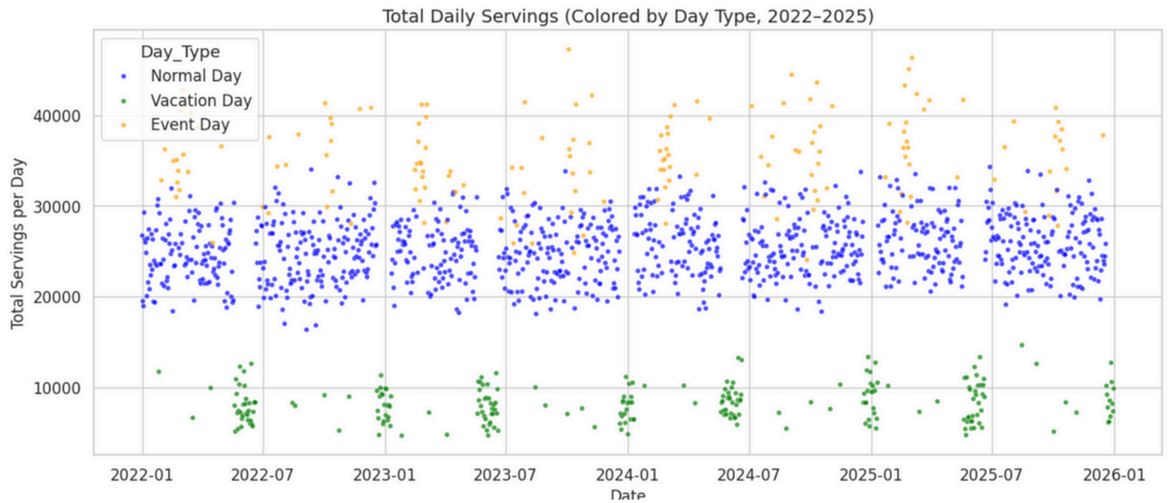
Converted categorical fields into machine-readable codes:

- Meal_Type_Code
- Dish_Code
- Day_Code

Encoders saved for reuse during forecasting.

Target variable: Servings

EDA & KEY OBSERVATIONS

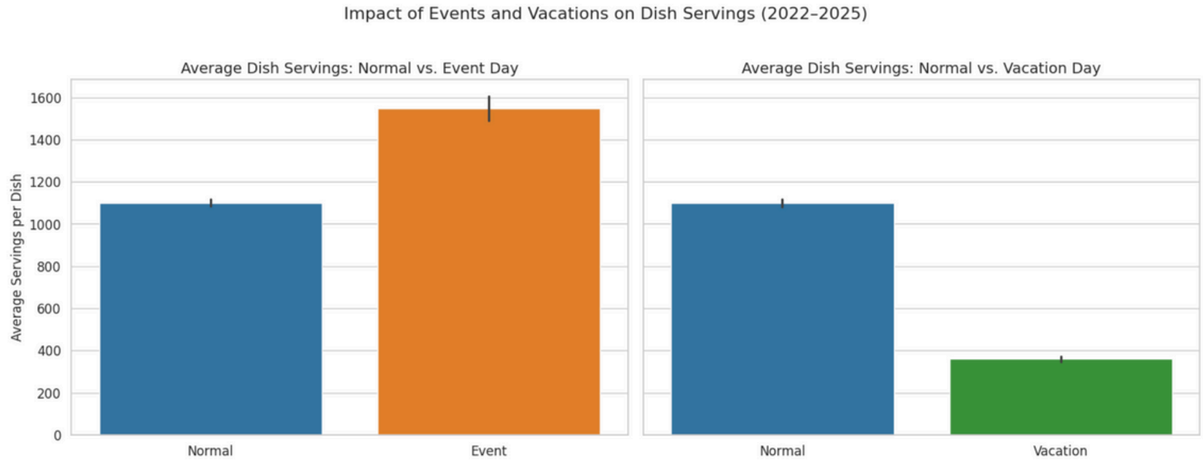


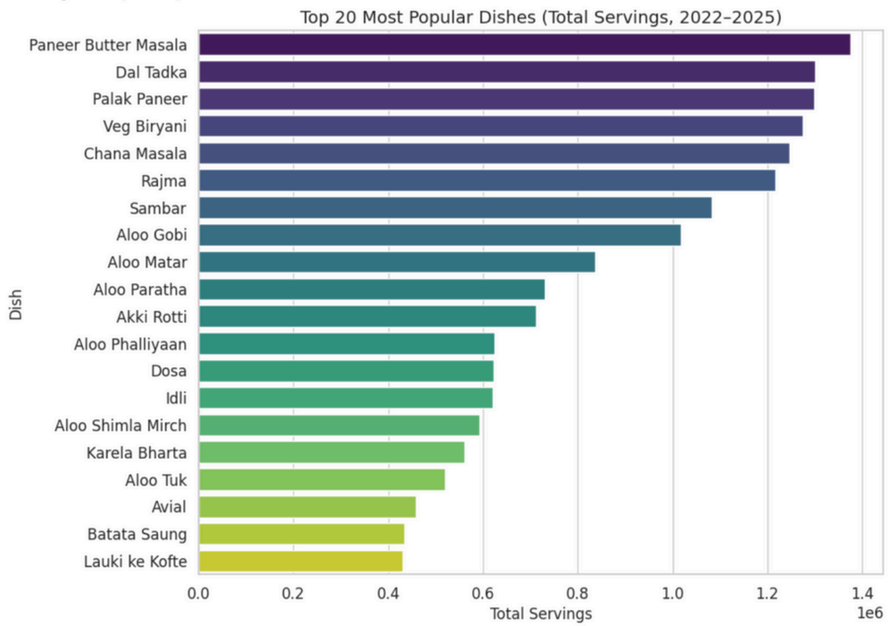
Total Daily Servings Trend

- Plot shows strong daily fluctuations in total servings over 4 years.
- Clear weekly seasonality: some days (Mon/Fri) consistently higher; weekends lower.
- Long-term trend rises slightly due to increasing Student_Count.
- Insight: Daily-level patterns justify using time-series models like Prophet and lag features.

2. Event & Vacation Impact

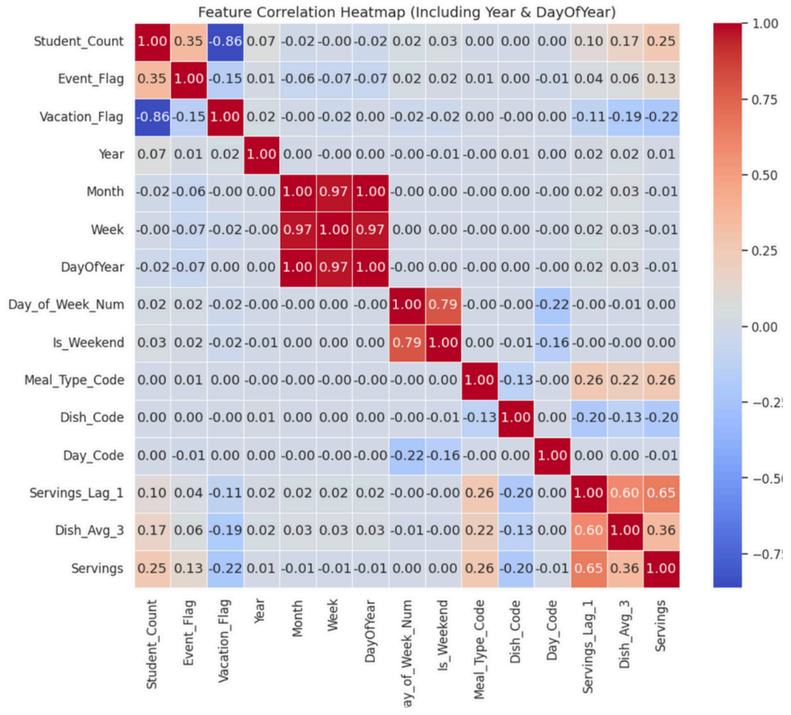
- Event Days → Sharp spike in demand
- Vacation Days → Heavy drop in demand
- Confirms that external factors (sports day, festivals, semester breaks) strongly influence food consumption.





Dish Popularity

- Top-20 dishes (e.g., Chana Masala, Rajma, Paneer dishes) dominate overall servings.
- Suggests which dishes students prefer the most → useful for menu planning and procurement.



Correlation Heatmap

- Servings_Lag_1 has the strongest positive correlation with today's demand.
- Student_Count also correlates with higher servings.
- Shows that time-dependent features are essential for accurate forecasting.

BASELINE ML MODELS & EVALUATION SETUP

1.Time-aware split: Data is split chronologically (first 80% train, last 20% test), so models are always predicting future days from past days and no data leakage occurs.

2. Models Compared (Baselines)

- Linear Regression – simple, assumes linear relationships.
- Random Forest Regressor – ensemble of decision trees, good for non-linear patterns.
- XGBoost (default / baseline) – optimized gradient boosting.

3. Metrics on test set:

- MAE – average absolute error.
- RMSE – penalizes larger errors more; used to pick the best model.
- MAPE – percentage error (computed only where Servings > 1).

Post-processing: negative predictions (if any) were clipped to 0, since negative servings are impossible.

Baseline Model Evaluation Completed!

	MAE	RMSE	MAPE
Random Forest	465.04	727.34	76.59
XGBoost	458.58	736.33	67.25
Gradient Boosting	481.60	767.93	77.33
Linear Regression	592.26	944.68	98.40

4. Evaluation:- Random Forest achieves the lowest RMSE (~727)

5. Why Random Forest is best baseline?

- Handles non-linear relationships between features (e.g., lag + student count + event flag).
- Naturally captures interactions (e.g., dish + meal type + weekday).
- More robust than Linear Regression and slightly better than default XGBoost before tuning

MODEL TUNING AND FINAL MODEL SELECTION

1.Hyperparameter Tuning Setup

- Used RandomizedSearchCV + TimeSeriesSplit (3 folds)
- → Ensures no data leakage and respects chronological order.
- Tuned both Random Forest and XGBoost on the training set.

2. Evaluation Method

- Both tuned models were tested on the same 20% hold-out test window.
- Used MAE, RMSE, and MAPE as metrics.
- Compared tuned results against the best baseline (Random Forest).

	MAE	RMSE	MAPE
Tuned XGBoost	443.28	704.80	66.79
Tuned Random Forest	454.35	722.49	71.31
Baseline (Random Forest)	465.04	727.34	76.59

3. Results — Tuned XGBoost Wins

- Tuned XGBoost achieved the lowest RMSE (~705)
- → Clearly better than both the baseline RF (~727) and tuned RF (~722).
- Shows improved prediction accuracy across all metrics.

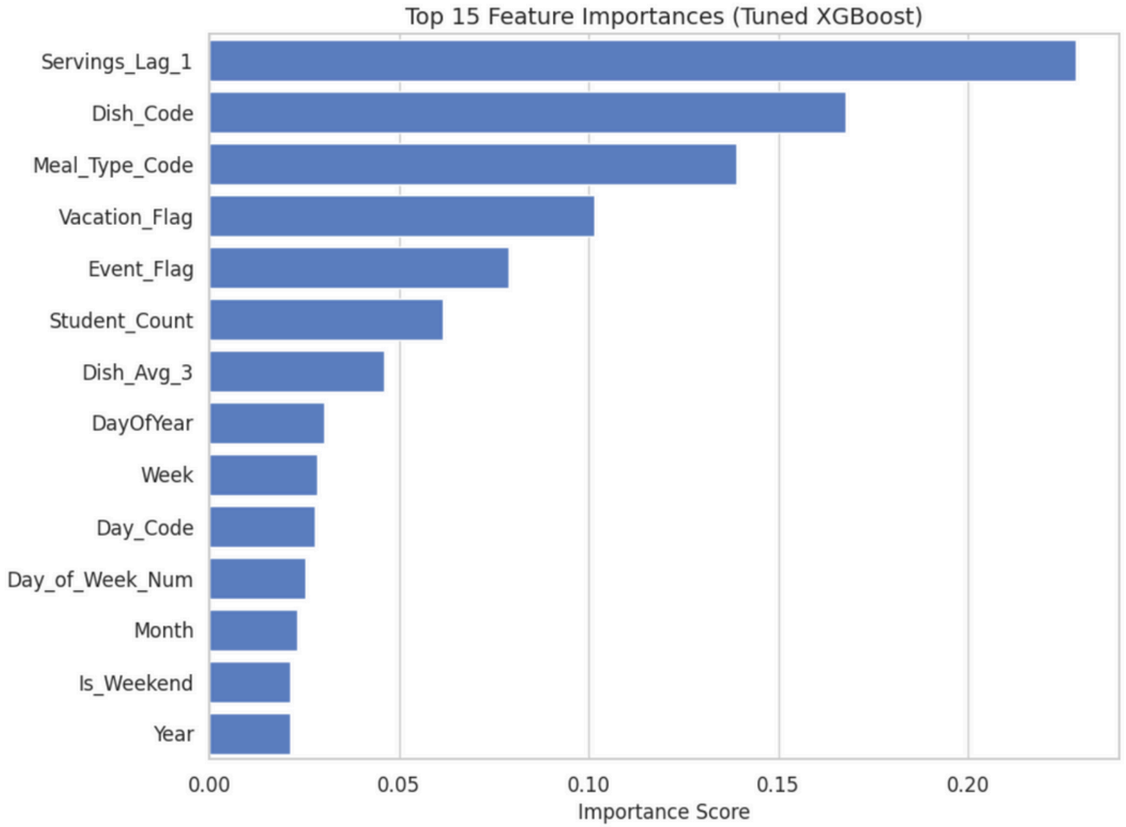
(You will show the “Final Model Comparison” bar chart here.)

4. Why Tuned XGBoost Works Best

- Learns non-linear patterns in demand.
- Handles interactions between temporal, categorical, and lag features effectively.
- More robust for large diverse dataset with many dishes.

Results After Model Tuning

- Uses **RandomizedSearchCV** with **TimeSeriesSplit** (3 folds) to tune **Random Forest** and **XGBoost** while strictly respecting time order (no data leakage).
- Evaluates tuned models on the same **20%** test window using **MAE, RMSE, and MAPE**, then compares them with the best baseline from Cell 6.
- Result: Tuned **XGBoost** becomes the final model with the **lowest RMSE (~705)** and improved error metrics over the baseline Random Forest.
- Saves this model as f**inal_best_model.pkl** and plots:
  - Final Model Comparison (RMSE across Tuned XGBoost, Tuned RF, and Baseline RF).
  - **Top 15 Feature Importances**, confirming Servings_Lag_1, Dish_Code, Meal_Type_Code, and event/vacation flags as the key drivers of predicted demand.



MODEL EXPLAINABILITY WITH SHAP (FINAL BEST MODEL)

1. What We Did

- Loaded the final_best_model.pkl (Tuned XGBoost).
- Used the same 80/20 time-series split.
- Took a 2,000-row sample from the test set for efficient SHAP computation.
- Computed SHAP values to measure each feature’s impact on predictions.

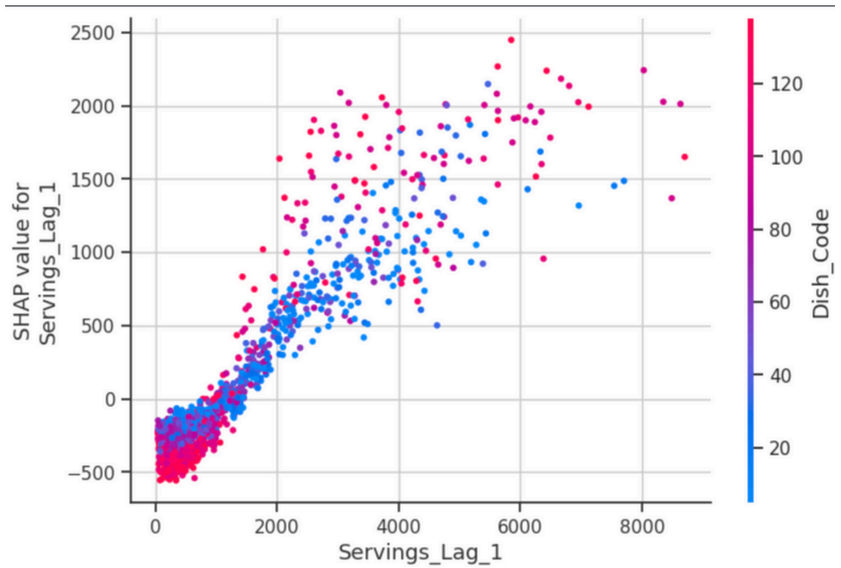
2.

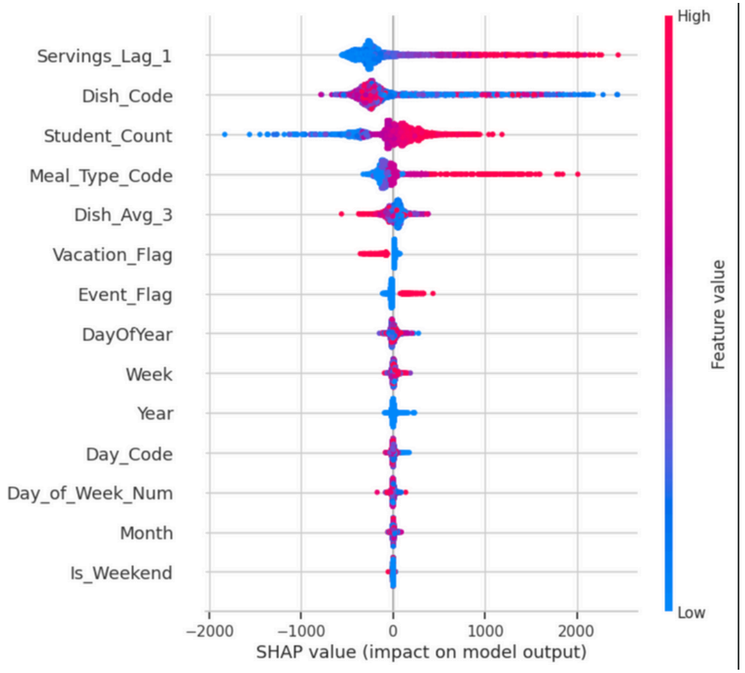
Interpretation:

Pink = high feature value pushing prediction up.

Blue = low feature value pulling prediction down.

The model's behavior is consistent and aligned with real cafeteria patterns.





Global Feature Importance (SHAP Summary Plot)

- The summary plot clearly highlights which features influence the model the most:
- Servings_Lag_1 → Strongest predictor
- Higher yesterday → Higher today (pink dots on right side).
- Dish_Code & Meal_Type_Code → Identify which dishes naturally have higher demand.
- Student_Count → More students = higher demand.
- Event_Flag & Vacation_Flag → Capture demand spikes/drops.

MODEL EXPLAINABILITY WITH SHAP (FINAL BEST MODEL)

3. Why Servings_Lag_1 Dominates

- When yesterday's servings were high, SHAP values are high → model predicts high demand.
- The trend is stable across all dish types (color variation = Dish_Code).

This proves the model is learning meaningful time-series behavior: popular dishes stay popular, and demand has momentum.

4. Why SHAP Matters

- Gives explainable AI—not a black box.
- Increases trust for cafeteria managers.
- Helps justify operational decisions:
  - Higher production on historically popular dishes
  - Preparing special menus on event days
  - Reducing waste on vacation days

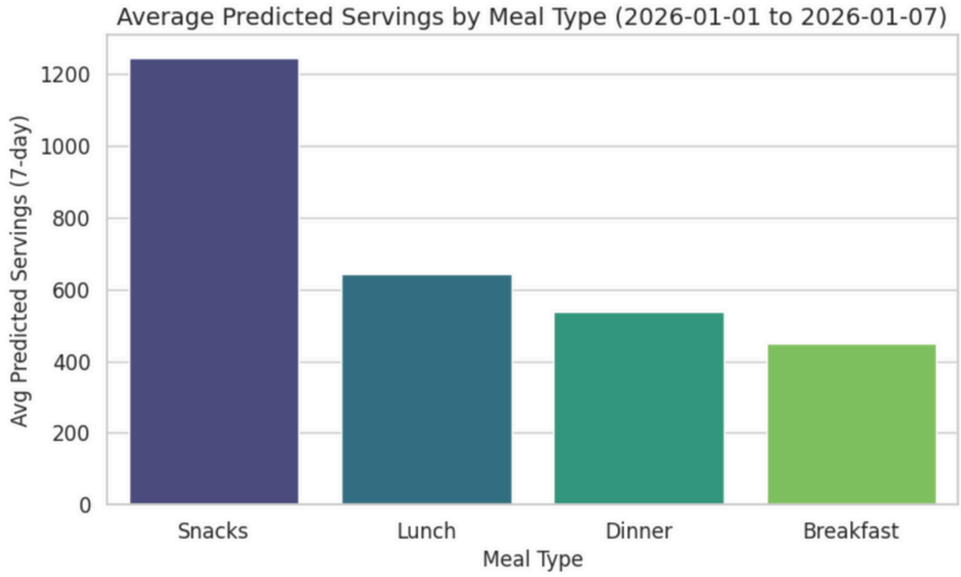
FUTURE FORECAST (DISH-LEVEL)

Goal: Predict cafeteria demand 7 days into the future for every dish meal combination using the Final Tuned XGBoost model.

What This Step Does

- Loads the final tuned XGBoost model + all saved label encoders and feature transformers.
- Ensures that future data is processed with the exact same pipeline as the training data.

	Date	Day	Meal_Type	Dish_Name	Predicted_Servings
0	2026-01-01	Thursday	Breakfast	Pongal	965
1	2026-01-01	Thursday	Breakfast	Akki Rotti	1549
2	2026-01-01	Thursday	Breakfast	Upma	188
3	2026-01-01	Thursday	Breakfast	Poha	405
4	2026-01-01	Thursday	Breakfast	Idli	902
5	2026-01-01	Thursday	Breakfast	Uttapam	1333
6	2026-01-01	Thursday	Breakfast	Dibba Rotti	338



PREFERENCE ANALYSIS (ASSOCIATION RULE MINING PER MEAL)

Goal:
Identify dish combinations that students frequently choose together — enabling better menu planning, combo offers, and serving layout optimization.
What This Step Does

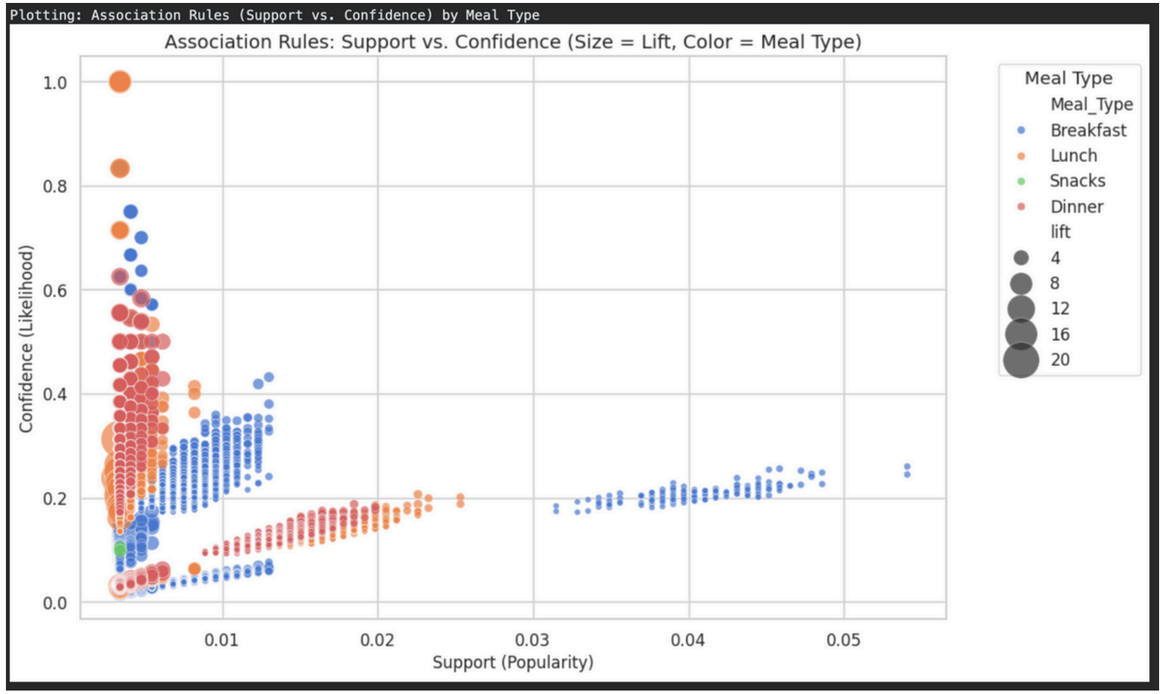
- Runs Apriori separately for Breakfast, Lunch, Snacks, Dinner.
- Treats each date × meal as a transaction basket of dishes served.
- Extracts rules like:
- “If students take Dish A, they often also take Dish B.”

Each rule is evaluated using:

- Support – how frequently the dish combination appears.
- Confidence – probability of choosing B when A is chosen.
- Lift (>1.0) – how much stronger the combination is compared to random chance.

Key Insights from Association Rules

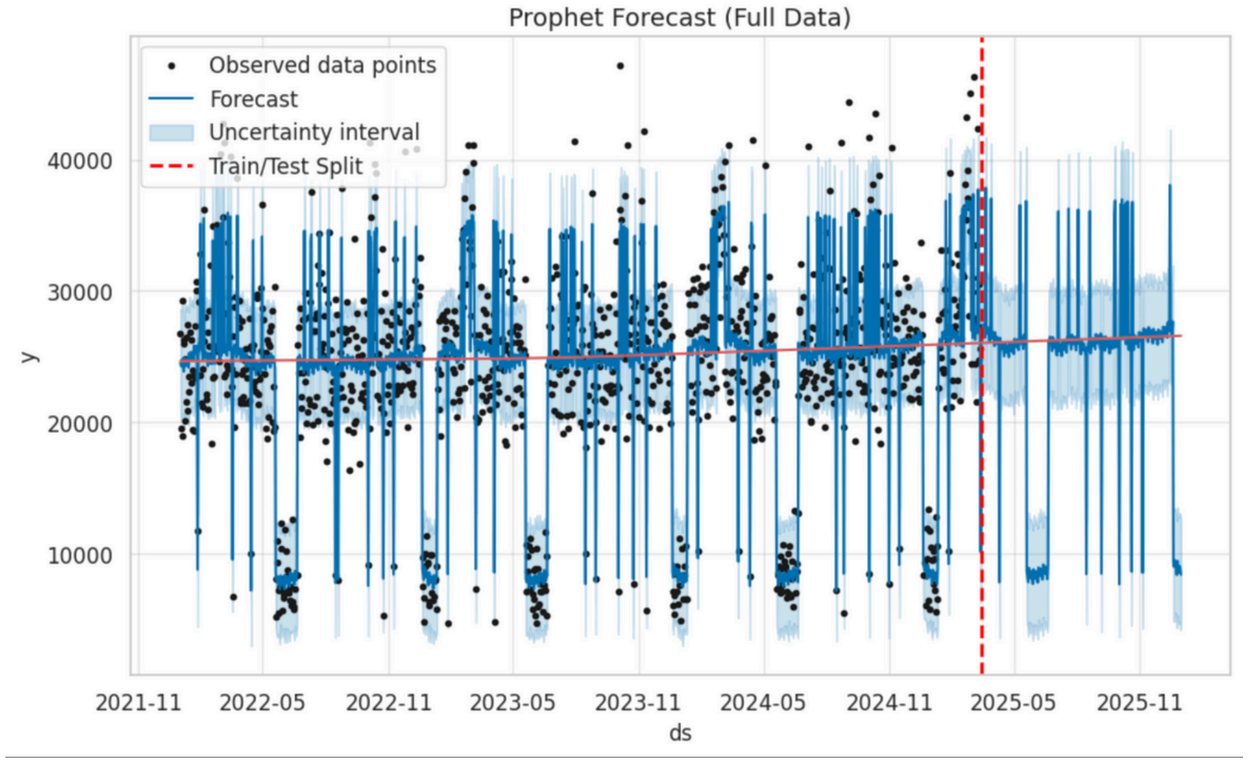
- Over 30,000 valid rules discovered across all four meals.
- Breakfast and Lunch show the strongest multi-item patterns due to greater dish variety.
- Many rules show very high lift (10–20×) → these dishes strongly “go together.”
- These patterns help cafeteria managers:
- Plan combos
- Place complementary dishes near each other
- Predict rush on specific items



Prophet Forecast vs. **Actual**

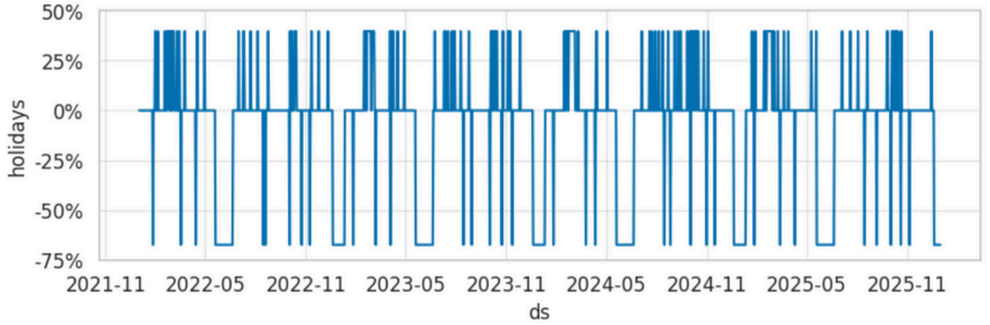
Prophet Model Evaluation (Aggregate Daily Servings)
MAE: 2579.93
RMSE: 3136.49
MAPE: 13.06%

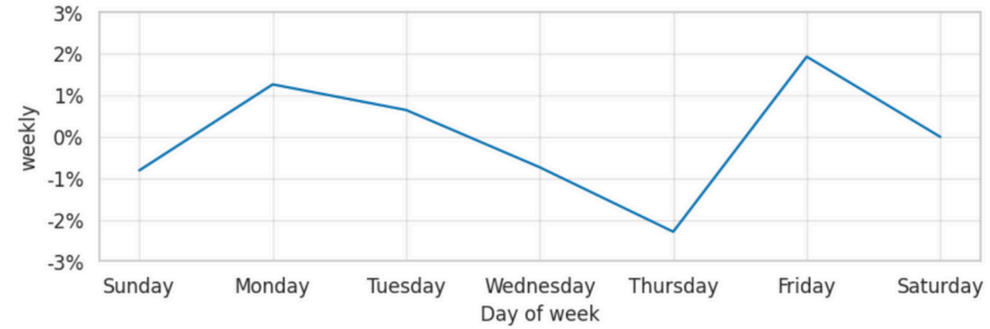
- Aggregates data to daily level and prepares the required **Prophet** format (ds, y).
- Adds **Event** & **Vacation** days as "**holidays**," enabling Prophet to model their positive/negative impact.
- Uses an **80/20 time-based split** to ensure correct forecasting evaluation.
- Fits the Prophet model and generates predictions for the test period.

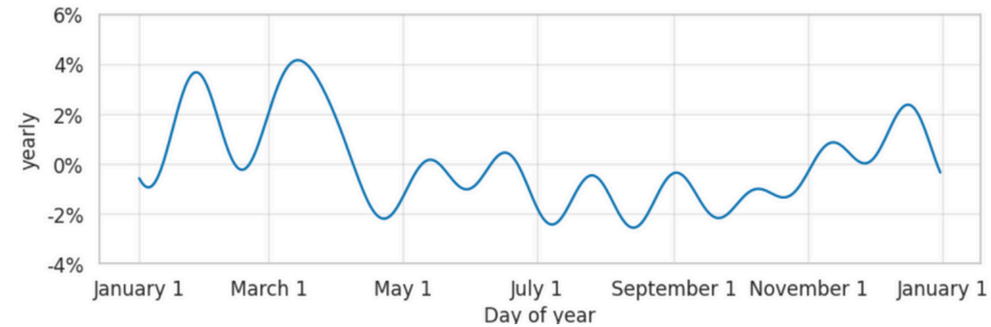


Overall Trend & Seasonality of the entire system.

- RMSE ≈ 3136 and MAPE ≈ 13% show a reasonable high-level forecast for total daily demand.
- Forecast Plot: Prophet captures long-term growth and repeated seasonal cycles across years.
- Component Breakdown:
- Trend: Shows steady demand growth over time.
- Holiday Effect: Events raise demand; vacations reduce it.
- Weekly Seasonality: Demand peaks on Monday & Friday, dips mid-week.
- This analysis validates the behavior of the full cafeteria system and provides strong explainability for decision-makers.







LIMITATIONS & FUTURE WORK

Limitations

- Synthetic dataset only; not yet validated against real cafeteria POS logs.
- No deep learning models (LSTM / Temporal CNN) even though proposed—only classical ML + Prophet used.
- Forecast assumes constant student count and no new events, which is unrealistic for live deployment.
- Synthetic generator simplifies human behavior; real cafeterias may have more complex eating patterns.

Future Work

- Integrate real POS/billing data and retrain the full pipeline.
- Implement LSTM / Temporal CNN for capturing long-range dependencies.
- Build the Streamlit dashboard for interactive forecasting & menu insights.

THANK YOU