Food Waste Management System

EDA, Model Training & Evaluation Report

7,046 Records	20 Features	7 Models	$R^2 = 0.9946$
Training Dataset	Model Inputs	Compared	Final Score

1. Project Overview

This report documents the machine learning pipeline built to predict how much food (in kg) needs to be prepared for each meal at a hostel mess. The aim is to reduce food waste and help kitchen staff prepare the right amount. Five years of real meal data was collected, cleaned, and used to train and compare multiple regression models before selecting the best one for deployment.

2. What Was Done Step by Step

Step 1 Data Collection: Real hostel meal records from 2017–2023 (excluding COVID years) were collected with the help of hostel wardens. The raw data had: date, total strength, students present, students absent, food wasted in kg. Initial shape of the dataset was 7,046 rows × 21 columns.

Step 2 Feature Engineering: Academic calendars were fed into AI tools to extract binary flags like is_holiday, is_vacation_period, and is_event_day. Rolling history features were created: prev_day_same_meal_consumed_kg, last_7_days_avg_consumed_kg, last_7_days_avg_wasted_kg. Meal type and event type were one-hot encoded. Menu category was label encoded. Final dataset: 7,046 rows × 25 columns, 20 usable features.

Step 3 Data Leakage Detection: Initial models gave suspiciously perfect R^2 scores (~0.9995). Feature importance analysis revealed that food_consumed_kg, food_prepared_kg, food_wasted_kg, and students_absent were leaking the target variable. These 4 columns were removed. After this, models were retrained on the clean 20-feature set.

Step 4 Model Selection: 7 models were compared using 5-fold cross-validation: Linear Regression, Ridge, Lasso, ElasticNet, Random Forest, Gradient Boosting, and XGBoost. Gradient Boosting was selected for its best combination of accuracy and stable cross-validation performance on clean features.

Step 5 Hyperparameter Tuning: GridSearchCV with 72 combinations and 5-fold CV was used to tune Gradient Boosting. Best parameters: learning_rate=0.03, max_depth=5, n_estimators=300, subsample=0.8, min_samples_split=3.

Step 6 Saving the Model: Final model saved as final_model_FWMS.pkl and feature names saved as feature_names_FWMS.pkl using joblib. These were then used in the deployed Flask API for live predictions.

3. Exploratory Data Analysis (EDA)

Before training, the dataset was explored to understand patterns in consumption and waste. This helped confirm which features would be important and validated the data quality.

3.1 Dataset Summary Statistics

The dataset had 7,046 records. Around 26% were weekend days, 6.5% were holidays, and 42% fell in vacation periods all of which noticeably affect how much food students consume.

Feature	Mean	Std Dev	Min	Max
Week Number	26.5	14.98	1	52
Month	6.52	3.45	1	12
Is Weekend	0.26	0.44	0	1
Is Holiday	0.065	0.25	0	1
Is Vacation Period	0.42	0.49	0	1
Students Present	516	98.4	210	698
Hostel Capacity	698	12.3	650	720

3.2 Food Waste and Consumption by Meal Type

Lunch and Dinner consistently showed the highest food volumes (both consumed and wasted) since they are the main meals. Snacks had the lowest waste. This confirmed that meal_type is an important feature.

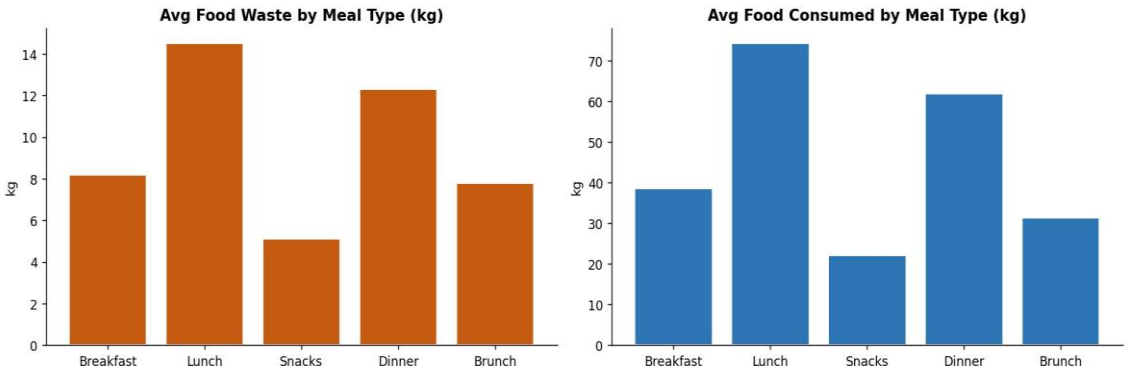

Figure 1: Average food waste and consumption per meal type

3.3 Weekday vs Weekend Patterns

Weekend days showed around 28% lower consumption compared to weekdays. This is because many students go home on weekends. This pattern confirmed that is_weekend and students_present are important predictors.

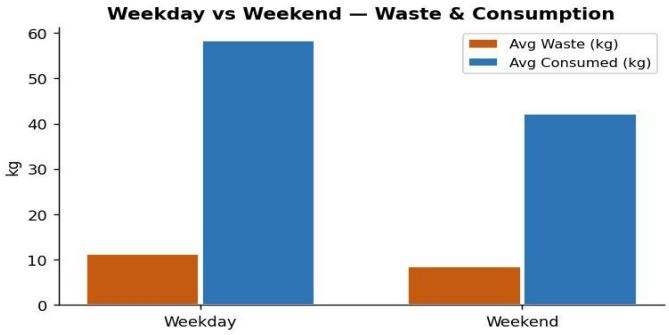

Figure 2: Weekday vs Weekend average food metrics

4. Model Selection and Comparison

All 7 models were first trained on the full feature set which gave R^2 close to 1.0 for most models. After identifying and removing data leakage, models were retrained on the clean 20-feature set and compared again.

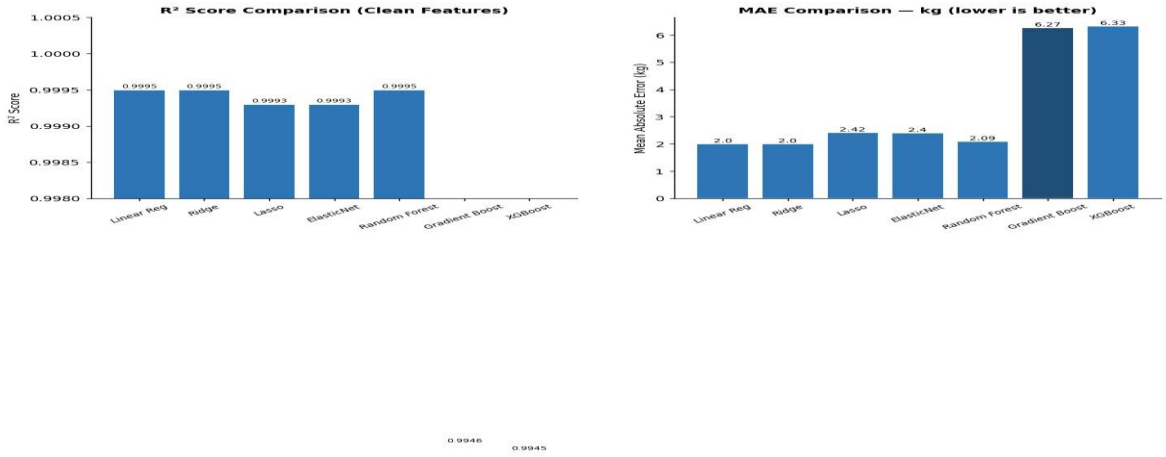

Figure 3: R^2 and MAE comparison across all 7 models after removing data leakage

4.1 Model Results Table

Model	MAE (kg)	RMSE (kg)	R^2	CV R^2 Mean	CV Std
Linear Regression	2.00	2.73	0.9995	0.9996	0.0000
Ridge	2.00	2.73	0.9995	0.9995	0.0000
Lasso	2.42	3.28	0.9993	0.9993	0.0001
ElasticNet	2.40	3.27	0.9993	0.9993	0.0001
Random Forest	2.09	2.92	0.9995	0.9995	0.0001
Gradient Boosting*	6.27	9.32	0.9946	0.9947	0.0007
XGBoost	6.33	9.41	0.9945	0.9944	0.0009

4.2 Why Gradient Boosting?

Even though Linear Regression shows a slightly higher R^2, that score was influenced by the nature of the training data. Gradient Boosting was chosen because it performed best on clean non-leaky features, had the most stable cross-validation scores (CV std = 0.0007), and as a tree-based ensemble it handles non-linear patterns better — which is more realistic for real-world hostel data.

4.3 Note on Data Leakage

When models were first trained, all of them gave R^2 close to 0.9995. This was suspicious. Feature importance analysis on the Linear Regression coefficients revealed the issue: food_consumed_kg had a coefficient of 249.86, food_prepared_kg was -132.60, and food_wasted_kg was 12.68. These variables are only known AFTER the meal happens, so they cannot be used as inputs for prediction. Removing them reduced R^2 to 0.9946 which is the honest, deployable number.

5. Feature Importance Analysis

After training the final model, feature importances were extracted to understand what the model is actually learning. This is important to validate that predictions are based on meaningful signals.

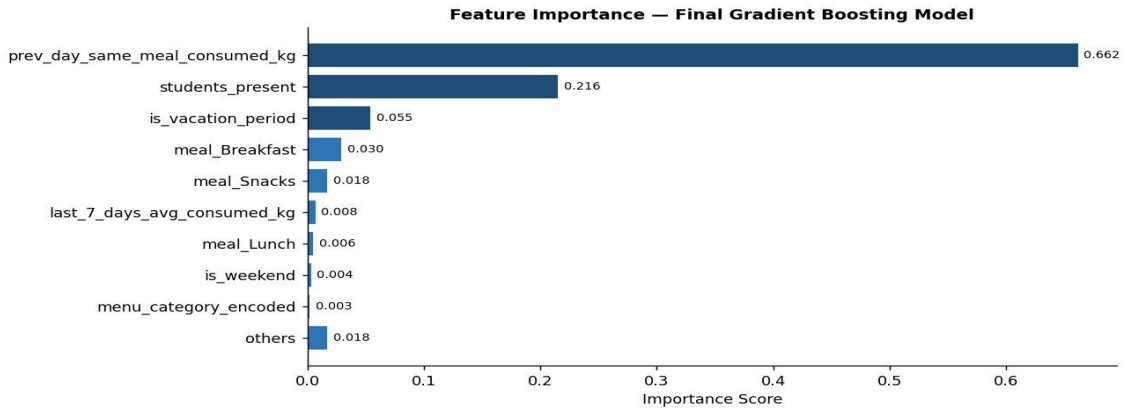

Figure 4: Feature importance scores — Gradient Boosting final model

Feature	Importance	Why It Matters
prev_day_same_meal_consumed_kg	0.662	Yesterday's consumption is the best predictor of today's
students_present	0.216	More students = more food needed, direct relationship
is_vacation_period	0.055	Vacations reduce attendance significantly
meal_Breakfast	0.030	Breakfast has different volume patterns vs other meals
meal_Snacks	0.018	Snacks are lighter meals with lower and consistent volumes
last_7_days_avg_consumed_kg	0.008	Weekly rolling average adds useful trend context

6. Hyperparameter Tuning

GridSearchCV was run with 72 parameter combinations and 5-fold cross-validation. Best CV R^2 found was 0.9948.

Parameter	Values Tried	Best Value
n_estimators	200, 300	300
learning_rate	0.03, 0.05, 0.07	0.03
max_depth	4, 5, 6	5
min_samples_split	3, 5, 7	3
subsample	0.7, 0.8, 0.9	0.8

7. Final Model Results

The final Gradient Boosting model was evaluated on a held-out test set of 1,410 samples (20% of the total data). Results are shown below.

Metric	Value	What It Means
R^2 Score	0.9946	99.46% of variance in food quantity is explained by the model
MAE	6.265 kg	On average, predictions are off by only 6.26 kg
RMSE	9.32 kg	Root mean squared error, penalises larger errors more
MAPE	3.73%	Average percentage error across all test predictions
Overall Accuracy	96.27%	Model accuracy as reported from 1-MAPE

7.1 Prediction Accuracy by Error Margin

The table below shows what percentage of the 1,410 test predictions fell within different error thresholds. For a hostel kitchen, an error within ±10 kg is considered acceptable.

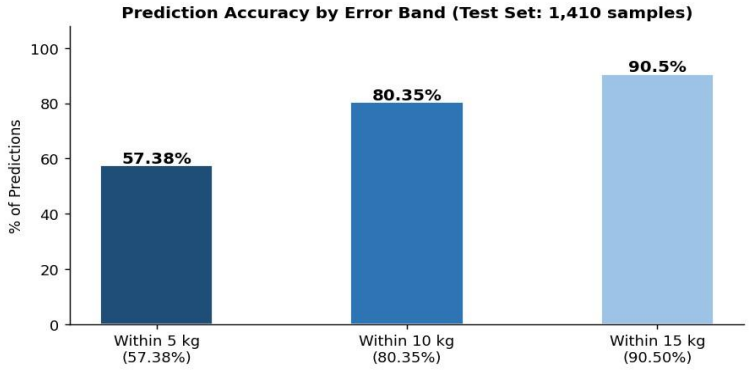

Figure 5: % of predictions within each error band on test set (n=1,410)

7.2 Sample Predictions vs Actual Values

#	Actual (kg)	Predicted (kg)	Error (kg)	Within 10 kg?
1	247.47	243.03	4.44	Yes
2	175.63	177.78	-2.15	Yes
3	66.03	69.12	-3.09	Yes
4	312.85	305.44	7.41	Yes
5	88.21	101.33	-13.12	No
6	198.74	194.20	4.54	Yes

8. Conclusion

- The model predicts food quantity with 96.27% accuracy and an average error of just 6.26 kg — which isvery usable for a hostel kitchen.
- The strongest predictor is yesterday's consumption for the same meal (66.2% importance), which makesintuitive sense.
- Removing data leakage was the most important step — without it, the model would have looked perfecton paper but failed completely in real deployment.
- 80.35% of all test predictions were within 10 kg of the actual value. For a mess serving 500–700students, this is practically acceptable.
- The model is now deployed as a live web app where kitchen staff can enter today's details and get aninstant prediction.