Ecommerce Data Science

This document details the thought and work behind the DS part of the project. The end goal here is to demonstrate our trained recommendation system is better than baseline heuristics.

Disclaimer: The food data is completely mocked up using the latest Gemini Pro model.
The labeled data for training is synthetically generated since no similar labeled dataset exists for training.
This is a Proof of Concept (POC). A production version would require clinical dataset validation and rigorous R&D.

Phase 1: Synthetic data generation

Since it's impossible for this POC to work on real data, our strategy is having a sophisticated “oracle” formula that simulates reality, generating a decent amount of cases from it and training the model.
The goal of our model will be to learn this formula without being told what it is.
We generated 50k entries from that oracleformula.
Notable properties we are looking for learning:

$$\text{Risk} = \underbrace{(\text{Carbs} + 1.5 \times \text{Sugar})}_{\text{Fuel}} \times \underbrace{(\frac{\text{GI}}{50})}_{\text{Speed}} \times \underbrace{\text{Multipliers}}_{\text{Context}}$$

The Hidden Multipliers (The "Gotchas"):

1. **Current Glucose > 160:** Risk $\times 1.5$
2. **Night Time:** Risk $\times 1.4$
3. **Pregnancy Week > 24:** Risk $\times 1.25$
4. **High Avg Glucose:** Risk $\times 1.2$
5. **Trend is Rising:** Risk $+(\text{Sugar} \times 2.0)$ *(Note: This is additive, not multiplicative!)*

Phase 2: Training the model

We treated this as a binary classification problem, if a food is safe or not given the circumstances.

The algorithm used is XGBoost Classifier (eXtreme Gradient Boosting).

Input Features is a 9-Vector containing:

User Context: glucose_level, glucose_trend, glucose_avg, pregnancy_week, time_of_day, intensity.

Food Attributes: food_carbs, food_sugar, glycemic_index.

Dataset: 50,000 synthesized rows (80/20 Train-Test split).

Training: using a 5-fold cross_val.

Results:

Added a sanity check to account for top 5 Most Important Features, just to verify we didnt get a high success rate as a false negative:

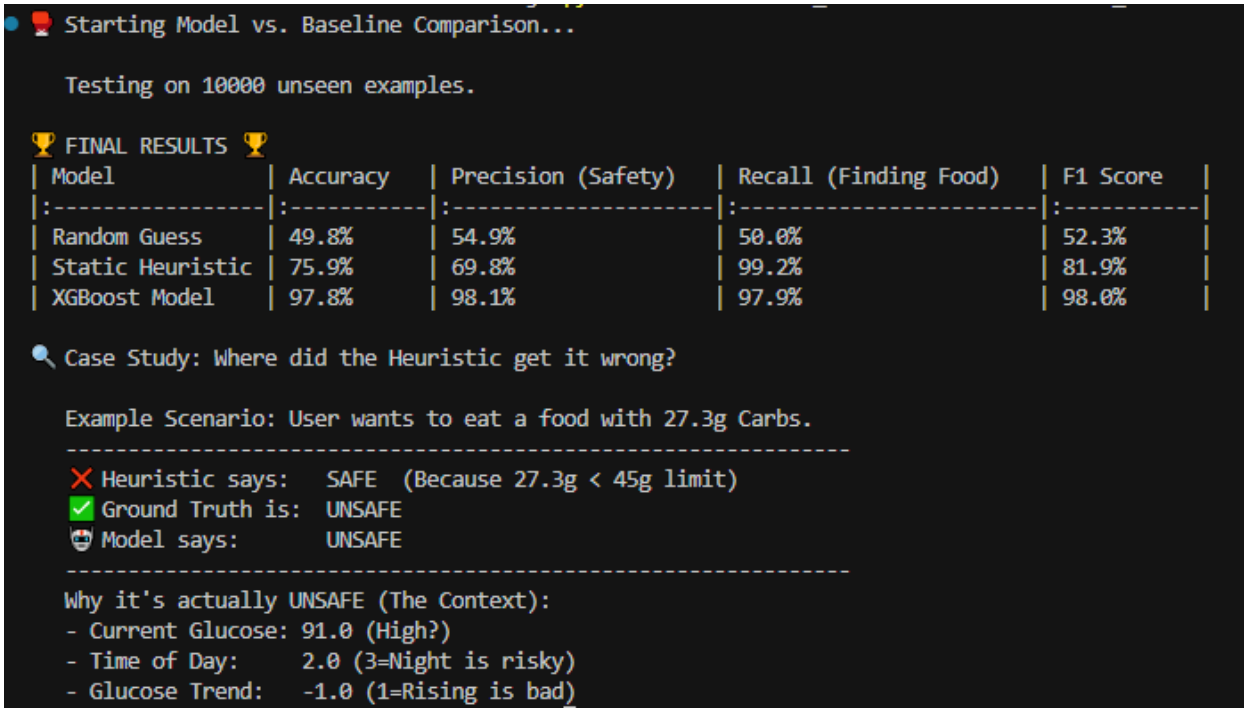
	Feature	Importance
7	food_carbs	0.520089
8	food_sugar	0.132825
2	glucose_trend	0.086624
5	time_of_day	0.067157
6	food_gi	0.059915

Which aligns with what we expected:

- Since carbs range from 0–100g, this number is the biggest contributor to the final score. The model learned that portion size matters most (which is true in the real world as well).

- Context is important: glucose trend and time of day amount for 15% of the importance, meaning the model learned that user state is important (correlating with night-time and rising trend penalties).
- Lower importance of food_gi: It's lower than carbs because GI is a multiplier, not a raw number. A high GI on a low-carb food (like a small candy) isn't as dangerous as a medium GI on a high-carb food (like a big bowl of pasta).

Phase 3: Compare against different baselines



We compared it against a random 50/50 and against a baseline "Static Heuristic" that mimics standard clinical advice (approving any food with <45g of carbohydrates).

Key Findings:

1. Better Precision: While the static heuristic achieved a baseline accuracy of 75.9%, it suffered from low precision (69.8%), meaning it frequently generated **False Positives** - identifying unsafe foods as safe.
2. Context Consideration: The XGBoost model achieved **97.8% accuracy** and **98.1% precision**, significantly reducing the risk of accidental glucose spikes.

3. Validation: As demonstrated in the case study, the model correctly rejected "moderate carb" foods (e.g., 27.3g) during high-risk windows (Evening / Night) that the static heuristic failed to flag. This confirms that giving user context (Time, Trend, Glucose) provides a measurable safety advantage over static nutritional rules.