Group 6

RESEARCH PAPER : NUTRIGEN

Personalized Meal Plan Generator Leveraging Large Language Models to Enhance Dietary and Nutritional Adherence

Authors : Saman Khamesian , Asiful Arefeen , Stephanie M. Carpenter , and Hassan Ghasemzadeh

Date : 11-01-2025

RESEARCH PAPER: NUTRIGEN

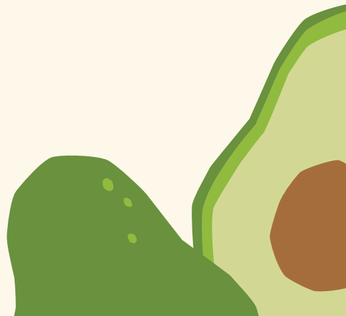
PERSONALIZED MEAL PLAN GENERATOR LEVERAGING LARGE LANGUAGE MODELS

NutriGen leverages LLMs to generate personalized meal plans using USDA nutritional data, achieving <4% error rates with minimal user input.

What are they solving in this framework?

- Alternative food suggestion
- Recipe Generation
- Ingredient availability

Core question: Can LLMs generate accurate, personalized meal plans that adhere to nutritional targets?



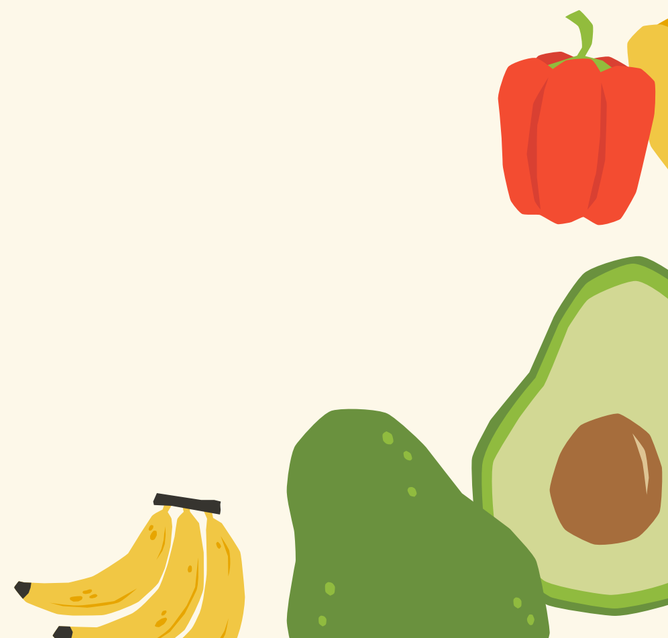
PREVIOUS WORKS

Existing Approaches & Limitations

1. IoMT-Assisted Systems (Iwendi et al.)
ML + medical sensors for dietary recommendations
✓ Highly precise, patient-specific
✗ Requires continuous medical monitoring, not for general population

2. ChatDiet (Yang et al.)
LLM-based chatbot for dietary advice
✓ Interactive and engaging
✗ High computational costs, scalability issues, platform dependencies

3. Particle Swarm Optimization (PSO) (Narendra et al.)
Optimization algorithms for nutritional balance
✓ Efficiently balances macronutrients
✗ Ignores user preferences, cultural restrictions, and dietary diversity



HOW WE APPROACHED THIS PAPER:

Iteration 1: First Reading (High-Level Understanding)

Questions We Had After Reading:

❓ How does PSO work in nutrition domain?
❓ Did they actually implement RAG or just discuss it?
❓ How did they validate LLM outputs against USDA?

Initial Confusion:

- Paper mentions "RAG framework" but methodology unclear
- Talks about "personalized database" but details missing
- References "dynamic retrieval" but process not explained

HOW WE APPROACHED THIS PAPER:

Iteration 2: Deep Dive (Finding Answers):

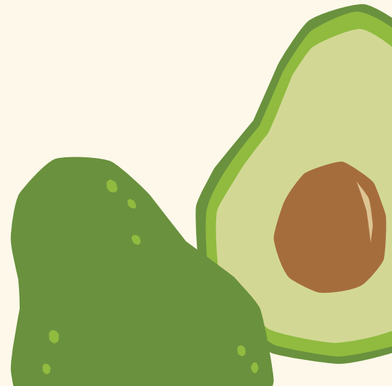
What We Discovered:

Q: Did they implement RAG?
A: NO – they discussed it as "future work"

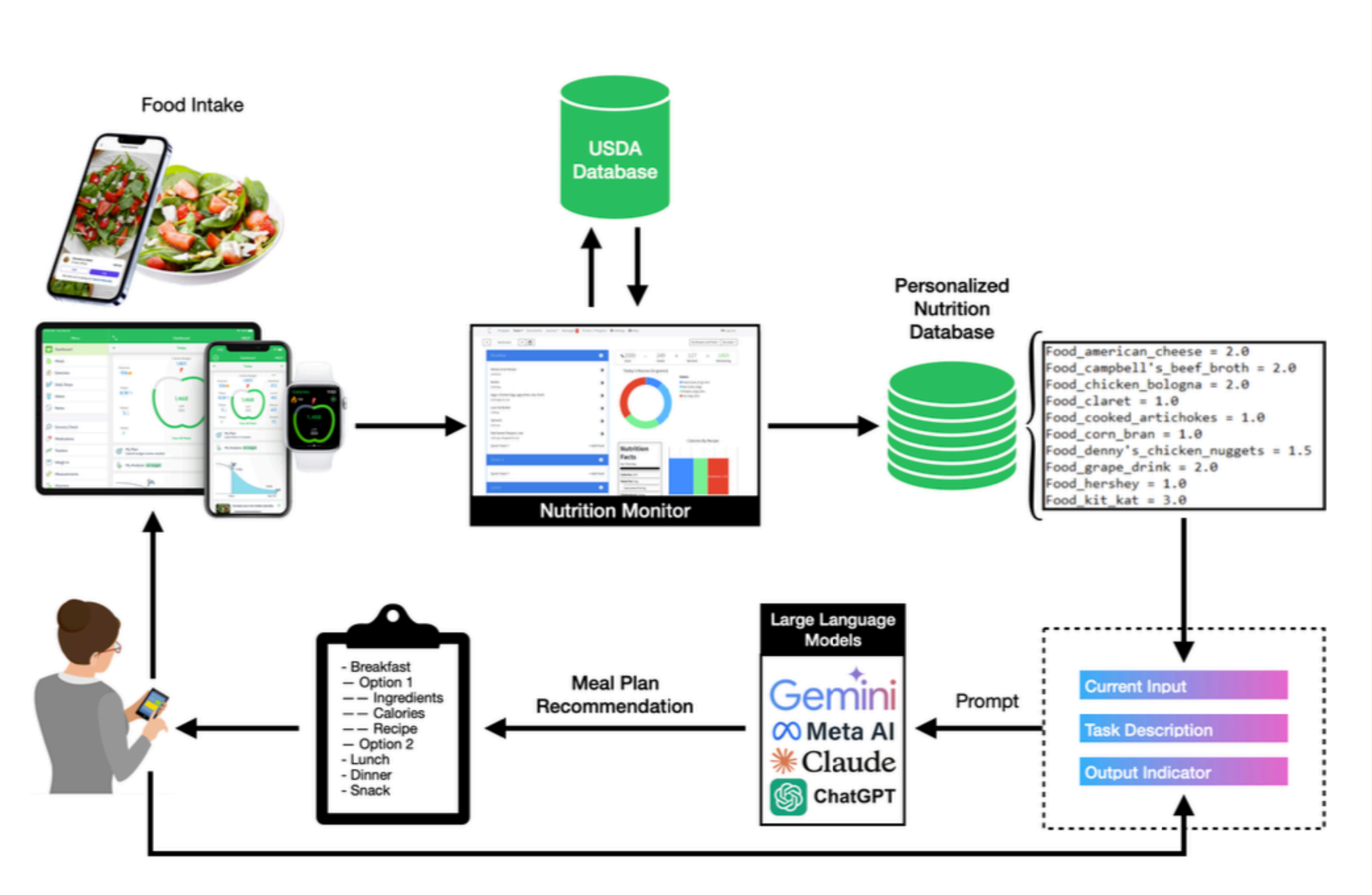
- Used static 200 foods embedded in prompt
- NOT dynamic retrieval
- Paper says: "RAG introduces latency... future work"

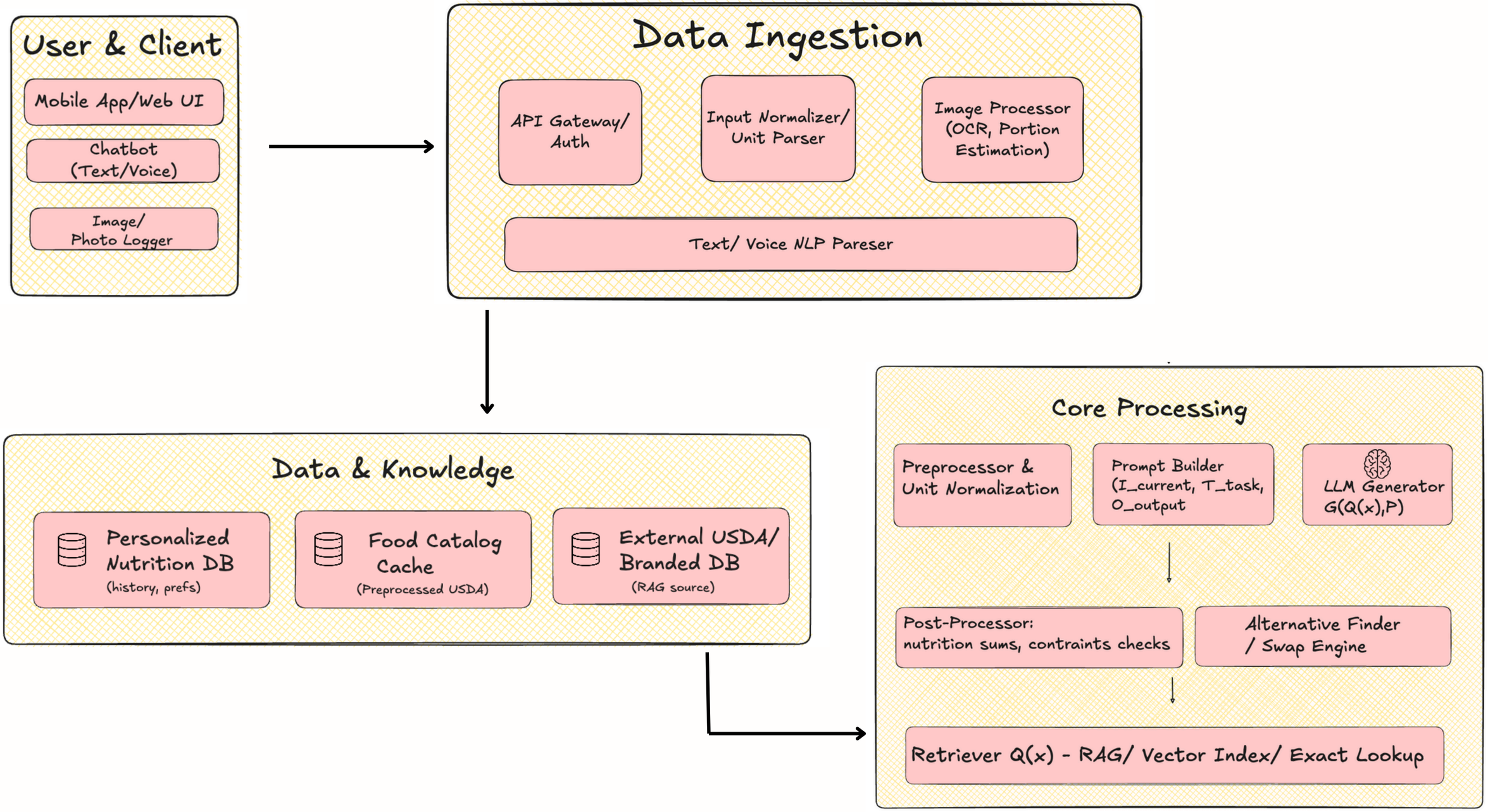
Q: How did they validate results?
A: Compared LLM outputs vs USDA nutritional values

- Measured if reported calories matched database
- Checked if total met user's target (2000 cal)
- 4 metrics: Accuracy, Speed, Target Hit, Completeness



PAPER ARCHITECURE DIAGRAM




User & Client
Mobile App/Web UI
Chatbot (Text/Voice)
Image/ Photo Logger
Data Ingestion
API Gateway/ Auth
Input Normalizer/ Unit Parser
Image Processor (OCR, Portion Estimation)
Text/ Voice NLP Pareser
Data & Knowledge
Personalized Nutrition DB
(history, prefs)
Food Catalog Cache
(Preprocessed USDA)
External USDA/ Branded DB
(RAG source)
Core Processing
Preprocessor & Unit Normalization
Prompt Builder (I_current, T_task, O_output
LLM Generator G(Q(x),P)
Post-Processor: nutrition sums, contraints checks
Alternative Finder / Swap Engine
Retriever Q(x) - RAG/ Vector Index/ Exact Lookup

WHAT DID NUTRIGEN ACTUALLY DO?

RESEARCH GOAL: TEST WHICH LLMS WORK BEST FOR MEAL PLANNING

What They Did: (Experimental Setup)

- Tested 10 different LLMs
- Used 200 foods from USDA (static dataset)
- Measured 4 metrics: Speed, Accuracy, Completeness, Target Hit
- Generated synthetic user profiles

What they **claim** they did:

Paper Claims:		Reality:
"RAG framework"	→	❌ "Future work"
"Dynamic retrieval"	→	❌ Static 200 foods
"Personalized database"	→	❌ One-time CSV

RESULTS FROM THE PAPER

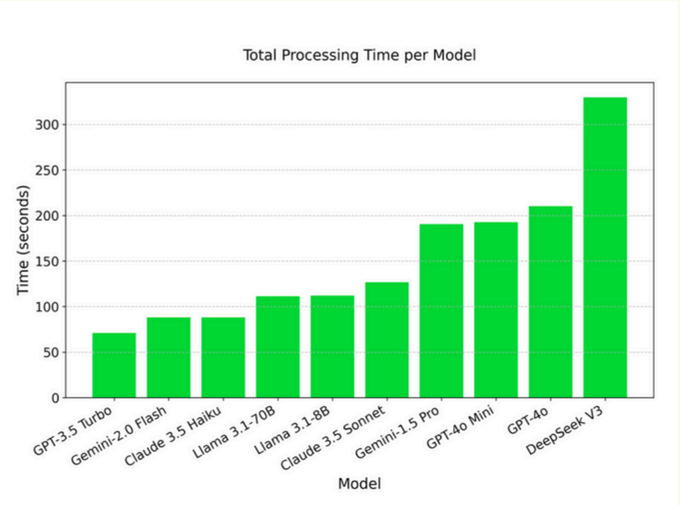
$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{1}{M}\sum_{j=1}^{M}\sum_{k=1}^{P} c_{\text{actual},i,j,k} - C_{\text{target},i}\right| \quad (2)$$

where:

- $N = 10$ represents the total number of inputs.
- $M = 3$ represents the number of meal plans per input.
- P represents the number of items in each meal plan.
- $c_{\text{actual},i,j,k}$ is the calorie content of the k-th food item in the j-th meal plan for input i.
- $C_{\text{target},i}$ is the target calorie value for input i.

MEAN ABSOLUTE AND PERCENTAGE ERROR BETWEEN TOTAL CALORIES AND TARGET FOR EACH MODEL

Model	MAE	MAE (%)
Claude 3.5 Haiku	128.23	8.99
Claude 3.5 Sonnet	99.16	4.85
DeepSeek V3	190.61	4.85
Gemini 1.5 Pro	182.16	10.44
Gemini 2 Flash	179.16	9.74
GPT-3.5 Turbo	54.16	3.68
GPT-4o	189.76	13.47
GPT-4o Mini	329.06	24.67
Llama 3.1 8B	**34.14**	**1.55**
Llama 3.1 70B	109.21	8.08



$$MAE = \frac{1}{M}\sum_{j=1}^{M}\left|\sum_{k=1}^{P} c_{\text{reported},j,k} - \sum_{k=1}^{P} c_{\text{USDA},j,k}\right|$$

EVALUATION OF NUTRITION FACTS GENERATED BY LLMS: THIS TABLE PRESENTS THE ERROR BETWEEN THE AVERAGE REPORTED TOTAL CALORIES AND THE USDA DATABASE FOR VARIOUS MEAL PLANS ACROSS DIFFERENT MODELS. BOLD NUMBERS INDICATE THE MINIMUM ERROR, WHILE A DASH (-) SIGNIFIES THAT THE MODEL FAILED TO GENERATE A MEAL PLAN.

	Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7	Input 8	Input 9	Input 10
Claude 3.5 Haiku	570.69	124	88	103.5	-	75	76.75	100	83	113
Claude 3.5 Sonnet	138.09	124	88	105	96	75	77	96	83	113
DeepSeek V3	138.09	96.6	88	-	-	-	-	100	82.84	113
Gemini 1.5 Pro	588	173.59	87.75	153.5	445.5	75	76.75	96	83	52.59
Gemini 2 Flash	589.33	173.59	87.75	153.5	445.5	75	103.41	96	83	48
GPT-3.5 Turbo	138.09	73.59	37.75	53.5	**45.5**	**25**	**26.75**	46	**32.84**	62.59
GPT-4o	50	**24**	**38**	**45**	563	41.6	50	**36**	74.6	52.59
GPT-4o Mini	60.03	27.5	50	50	161.66	55	**26.75**	47.33	35	53.33
Llama 3.1 8B	88.09	58.33	50	60	46	58.33	**26.75**	-	50	**43.33**
Llama 3.1 70B	**45**	60	50	50	46.66	40	48.33	46	50	63

OUR IMPLEMENTATION & UNDERSTANDING

What we have done & Problems faced?

- Using the new LLM models : Gemini-2.5-pro, GPT-4, Llama 3.18b, GPT-3.5-turbo, Gemini-2.0-flash
- Implement the paper in the procedural way mentioned. (Target Vs Actual Calculation errors) (Solved through adding servings)
- Gemini models (Token & Prompt Warnings)(Solved by disabling the warnings & increasing the token limit)
- RAG Implementation (Defined in the future scope)

Understanding before & after Implementation

- Direct use of LLM without any limitations
- Execution Time
- Token Limitation
- API Rate Limit
- Agent Training

KEY TAKEAWAYS TO OUR PROJECT - MEALMIND

- Selection of LLMs
- Token Limitation
- Training the Agents
- Parallel Execution Architecture
- Servings Per Meal - Inventory
- Auto-Recovery Pipelines
- RAG
- How it effected our business approach to the project ?

THANK YOU