

Introduction:

For this assignment, we have chosen five packages of food items that will help a diet optimization model.

The items that I have chosen were Okios Yogurt, Avocado Toast, Chicken Fried Rice, Orange Chicken, and Mushrooms. Each item's nutrition facts have been analyzed based on the nutritional content, and photographs. With this, the items were also calculated for the cost per serving for each food item. This part of the step will be crucial as we will be laying down the groundwork to figure out the linear programming model which will help create an accurate cost assessment of the required nutritional values that were provided.

Linear Program Problem:

The Linear programming problem required several key aspects where decision variables, objective functions, and nutritional constraints will play a key part. The decision variables were defined as:

- okios_yogurt: Servings of Okios yogurt.
- a_toast: serving of avocado toast
- CFR: Servings of chicken fried rice
- OC: Servings of Orange Chicken
- mrooms: Serving of Mushrooms

I then applied an objective function that would help minimize the total cost dependent on the amount of servings.

Objective Function:

$$\text{Total_Cost} = 1.2475 * \text{okios_yogurt} + 3.59 * \text{a_toast} + 2.66 * \text{CFR} + 2.66 * \text{OC} + 1.2475 * \text{mrooms}$$

Furthermore, nutritional constraints were added to find the minimum or maximum requirements for sodium, energy, protein, vitamin D, calcium, iron, and potassium.

Linear Programming in Python:

The next part of this assignment was applying the Python PuLP library where it will be able to create the problem statement, objective function, and the nutritional constraints in code. With this code, we were able to execute it to find the optimal amount of servings for each of the food items while also calculating the minimum cost. The minimum total cost for a week when executing the code was \$276.76. With that, it led to 216 serving sizes of yogurt because due to all the sources that I provided, not all items had the nutritional facts whereas the yogurt did, showing that most items were not populated.

- Optimal Servings:
- Avocado_Toast: 0.00 servings
- Chicken_Fried_Rice: 0.00 servings
- Mushrooms: 0.00 servings
- Okios_Yogurt: 216.06 servings
- Orange_Chicken: 2.71 servings

- Total Cost: \$276.75

The problem that I encountered very early was the amount of vitamin D each item had. The only item that I listed that had vitamin D was the Okios Protein yogurt. Due to this, it caused a lack of variety in the program, which provided many serving sizes for the protein yogurt. Furthermore, constraints for the minimum or maximum for each of the food items were not added at this time. I then applied the yogurt maximum constraint, which was limited to 70 to satisfy the vitamin D constraint. Another aspect that was also difficult was that mushrooms in general didn't offer much to the nutritional constraints, which is why it mainly showed zero. But with the addition of the constraints, it changed drastically. The revised model produced an infeasible solution. This means that the constraints that I provided were too strict to try to satisfy the requirements for nutrition. But with the added constraints it allowed for other food items to be also in the equation where sometimes they were limited. With the added constraints, I would have to spend an extra \$116. The output that we received was:

- **Status:** Infeasible
- **Avocado Toast:** 15.0 servings
- **Chicken Fried Rice:** 92.364 servings
- **Mushrooms:** 0.0 servings
- **Okios Yogurt:** 70.0 servings
- **Orange Chicken:** 2.387 servings
- **Total Cost:** \$393.53

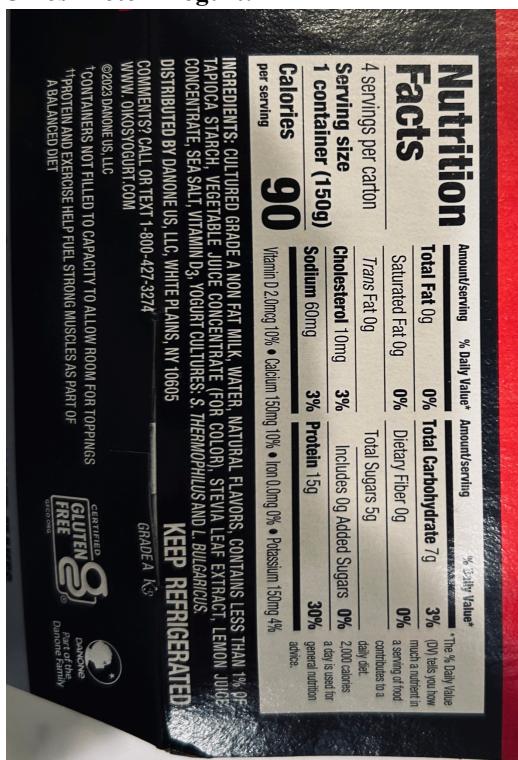
One aspect that I would introduce to my diet is fish and oatmeal since they contain a high amount of vitamin D. Also adding more food categories such as fruits, vegetables, and dairy items would be highly beneficial. This will add more constraints, which will be able to create more of a balance of items since five items of food is a small amount.

LLM:

The LLM agent that I will be using is [ChatGPT](#). I prompted a question to the LLM agent, “I am building out a Diet Plan model utilizing linear programming. We have built it out, and now we want to see if we can build an LLM model for The Diet Problem.” The agent was able to provide me with many different models but were more in the general aspect which didn't correlate much with the diet plan which means it needed to be more specific. I tried to be more specific on the needs of what I needed to be able to build an LLM based on the diet plan, and it didn't help in many ways, as it created more confusion. I then asked if it would be possible to write some code that would be able to help figure out a Diet plan using a gpt-3.5-turbo model to be able to figure out the best model that will adhere to the constraints. Furthermore, I think by using this, I would be able to create a great diet plan and be able to complete the assignment.

Appendix:

Okios Protein Yogurt:



Avocado Toast:

Serving Size: 1 pack

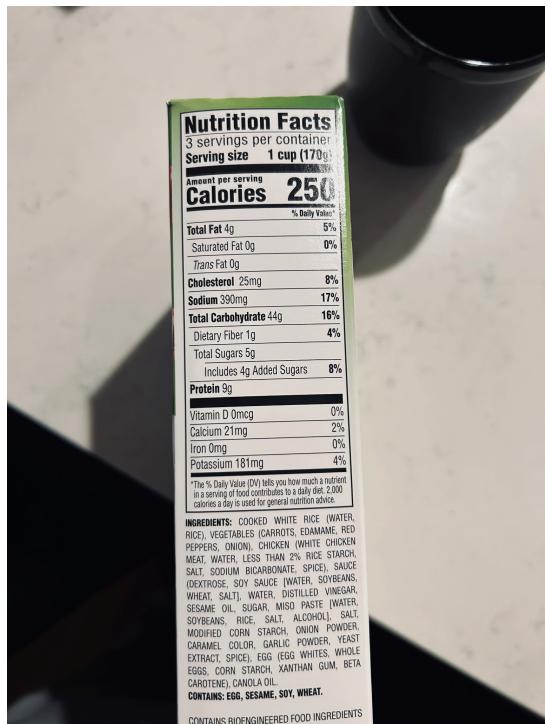
Serving Per Container: 1

Amount 1 pack (76g):

Calories: 190

	% Daily Value*
Total Fat 13gram	16%
Saturated Fat 2.5gram	13%
Trans Fat 0gram	0%
Cholesterol 0mg	0%
Sodium 350mg	15%
Total Carbohydrate 18gram	7%
Dietary Fiber 5gram	18%
Total Sugars 1gram	0%
Added Sugars 1gram	1%
Protein 4gram	0%

Chicken Fried Rice:



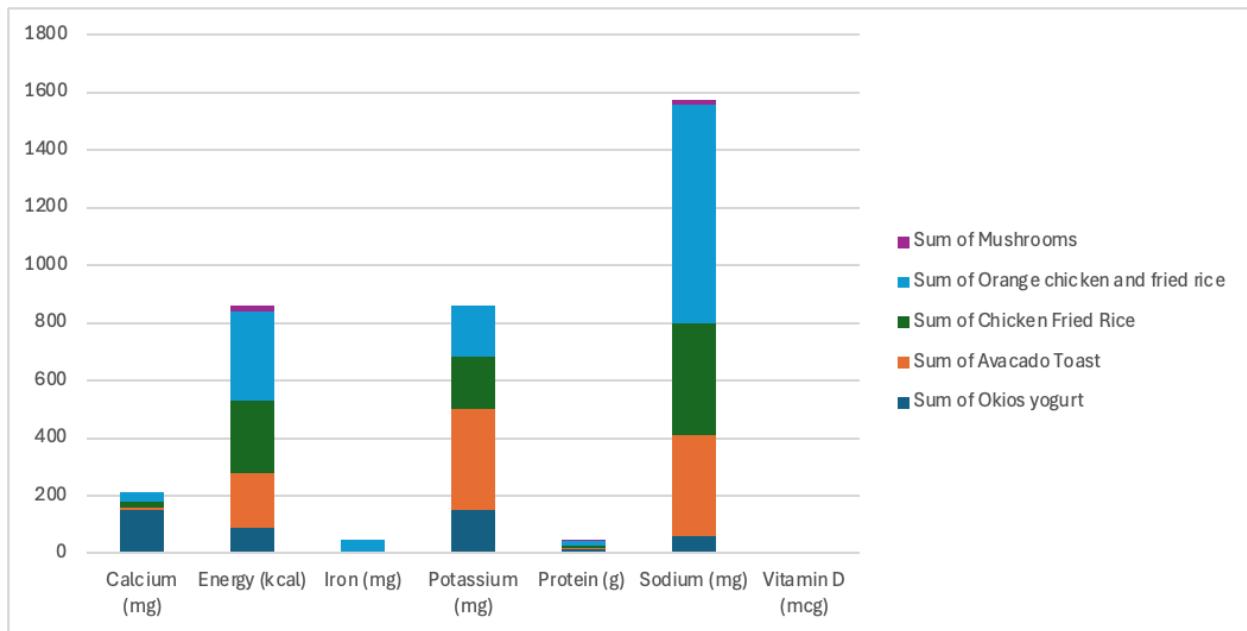
Orange Chicken:



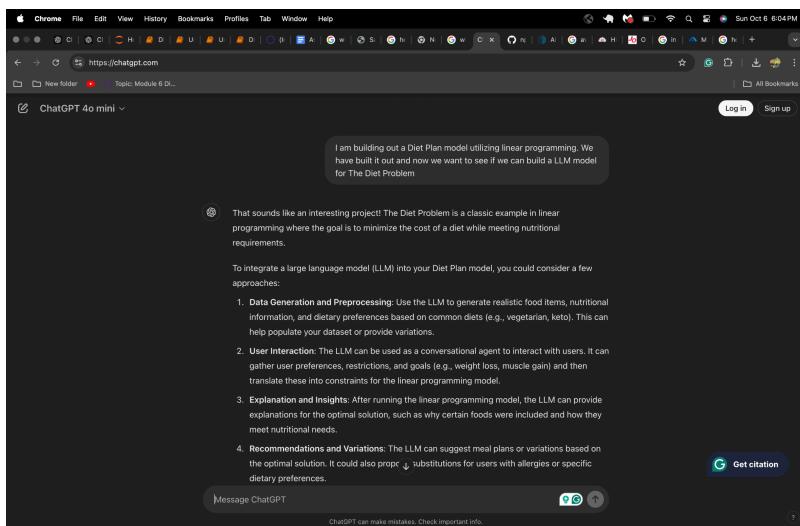
Mushrooms:

Nutrition Facts	
Serving Size :	3 Oz (65g)
Servings Per Container :	
Calories	20.0kcal
Total Fat	0.0g
Saturated Fat	0.0g
Trans Fat	0.0g
Cholesterol	0.0mg
Sodium	15.0mg
Total Carbohydrate	3.0g
Dietary Fiber	4.0g
Protein	8.0g
Vitamin A	0.0mcg
Vitamin C	0.0mg
Calcium	0.0mg
Iron	0.0mg

Component	Okios yogurt	Avacado Toast	Chicken Fried Rice	Orange chicken	Mushrooms
Sodium (mg)	60	350	390	760	15
Energy (kcal)	90	190	250	310	20
Protein (g)	15	4	9	13	8
Vitamin D (mcg)	2	0	0	0	0
Calcium (mg)	150	10	21	31	0
Iron (mg)	0	1	0	46.5	0
Potassium (mg)	150	350	181	181	0



LLM Portion:



I am building out a Diet Plan model utilizing linear programming. We have built it out and now we want to see if we can build a LLM model for The Diet Problem

That sounds like an interesting project! The Diet Problem is a classic example in linear programming where the goal is to minimize the cost of a diet while meeting nutritional requirements.

To integrate a large language model (LLM) into your Diet Plan model, you could consider a few approaches:

1. **Data Generation and Preprocessing:** Use the LLM to generate realistic food items, nutritional information, and dietary preferences based on common diets (e.g., vegetarian, keto). This can help populate your dataset or provide variations.
2. **User Interaction:** The LLM can be used as a conversational agent to interact with users. It can gather user preferences, restrictions, and goals (e.g., weight loss, muscle gain) and then translate these into constraints for the linear programming model.
3. **Explanation and Insights:** After running the linear programming model, the LLM can provide explanations for the optimal solution, such as why certain foods were included and how they meet nutritional needs.
4. **Recommendations and Variations:** The LLM can suggest meal plans or variations based on the optimal solution. It could also propose substitutions for users with allergies or specific dietary preferences.

5. **Feedback Loop:** Implement a system where users can provide feedback on the suggested diets, which the LLM can use to refine future recommendations.

Implementation Steps

1. **Define the Inputs:** Specify the input parameters the LLM will need, such as dietary restrictions, preferred cuisines, and nutritional goals.
2. **Model Interaction:** Create an interface where users can interact with the LLM to input their dietary preferences and constraints.
3. **Linear Programming Integration:** Connect the LLM's output to your linear programming model. Ensure that the LLM can generate constraints and objectives compatible with your model.
4. **Training and Fine-tuning:** If you have specific dietary knowledge or data, consider fine-tuning the LLM to better cater to your application's domain.
5. **Testing and Evaluation:** Evaluate the model's performance in generating valid diets and providing accurate nutritional information. User feedback can be critical in this phase.

By combining the strengths of linear programming with the adaptability of an LLM, you could create a robust tool for personalized diet planning. If you have any specific questions or areas you want to dive deeper into, let me know!

Vitamin D (mcg)

2

0

0

0

0

Calcium (mg)

150

10

21

31

0

Iron (mg)

0

1

0

46.5

0

Potassium (mg)

150

350

181

181

0

Here are my nutritional facts for each of these items that I have chosen what LLM model do you think would be sufficient

For your project involving the Diet Problem with specific food items and their nutritional facts, you have a couple of options when it comes to selecting an appropriate large language model (LLM).

Here are some recommendations based on different aspects of your needs:

Model Options

- GPT-3.5 or GPT-4:** These models are versatile and can handle a wide range of tasks, including generating diet plans, providing nutritional explanations, and creating user interactions. They are well-suited for understanding context and generating coherent responses.
- Fine-tuned LLM:** If you have specific dietary guidelines, recipes, or nutritional data, consider fine-tuning a model like GPT-2 or GPT-3 on that dataset. This can enhance the model's ability to understand and generate relevant content specific to your application.
- Domain-Specific Models:** Look for models that have been trained on health and nutrition data. While they may not be as general-purpose as GPT-3.5 or GPT-4, they could provide deeper insights into dietary guidelines and nutritional science.

Key Features to Consider

- Nutritional Knowledge:** Ensure the model has a good understanding of nutritional requirements and can generate meal plans that meet specific criteria (e.g., low sodium, high protein).
- Conversational Ability:** The model should be able to engage users in dialogue to gather preferences and provide explanations for meal choices.
- Flexibility:** The ability to adapt to various dietary restrictions (e.g., vegetarian, gluten-free) is

Implementation Steps

- Data Preparation:** Structure your data (nutritional facts) in a way that the LLM can easily reference. Consider using a tabular format or JSON.
- User Input Handling:** Design prompts that can effectively gather user dietary preferences, constraints, and goals.
- Response Generation:** Implement logic to connect user input with the dietary data and the linear programming model to generate recommendations.
- Feedback Loop:** Allow users to provide feedback on the generated meal plans so the model can learn and improve over time.

Would you be able to build out a LLM model for a diet plan

python

 Copy code

```
import openai

# Set up your OpenAI API key
openai.api_key = 'YOUR_API_KEY'

def generate_meal_plan(user_preferences):
    prompt = f"Generate a healthy meal plan for someone who is {' '.join(user_preferences)}"

    response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[{"role": "user", "content": prompt}],
        max_tokens=150
    )

    return response['choices'][0]['message']['content']

# Example user input
user_preferences = [
    "trying to lose weight",
    "vegetarian",
    "high protein"
]

meal_plan = generate_meal_plan(user_preferences)
print("Generated Meal Plan:")
print(meal_plan)
```

Please use these constraints and apply it towards this code

Component

Max/Min

Daily Amount and measure

Sodium

Maximum

5,000 milligrams (mg)

Energy

Minimum

2,000 Calories (kilocalories, kcal)

Protein

Minimum

50 grams (g)

Vitamin D

Minimum

↓
20 micrograms (mcg)

20 micrograms (mcg)

Calcium

Minimum

1,300 milligrams (mg)

Iron

Minimum

18 milligrams (mg)

Potassium

Minimum

4,700 milligrams (mg)



Certainly! Here's how you can modify the provided code to include the specified dietary constraints when generating a meal plan using OpenAI's API.



Code to Generate a Meal Plan with Specified Nutritional Constraints

```
python Copy code

import openai

# Set up your OpenAI API key
openai.api_key = 'YOUR_API_KEY'

def generate_meal_plan(user_preferences):
    # Define the nutritional constraints
    constraints = (
        "Please create a meal plan that adheres to the following nutritional constraints:\n"
        "1. Sodium: Maximum of 5,000 milligrams (mg)\n"
        "2. Energy: Minimum of 2,000 Calories (kcal)\n"
        "3. Protein: Minimum of 50 grams (g)\n"
        "4. Vitamin D: Minimum of 20 micrograms (mcg)\n"
        "5. Calcium: Minimum of 1,300 milligrams (mg)\n"
        "6. Iron: Minimum of 18 milligrams (mg)\n"
        "7. Potassium: Minimum of 4,700 milligrams (mg).\n"
    )

    # Create the prompt for the LLM
    prompt = f"Generate a healthy meal plan for someone who is {', '.join(user_preferences)}"

    response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[{"role": "user", "content": prompt}],
        max_tokens=250 # Increased token limit for more detailed responses
    )

    return response['choices'][0]['message']['content']
```

```
# Example user input
user_preferences = [
    "trying to lose weight",
    "vegetarian",
    "high protein"
]

meal_plan = generate_meal_plan(user_preferences)
print("Generated Meal Plan:")
print(meal_plan)
```