Model Development Phase Template

Date	11 July 2024
Team ID	SWTID1720188483
Project Title	Nutrition App Using Gemini Pro : Your Comprehensive Guide to Healthy Eating and Well-being
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:

```
meal_plans = {
    "breakfast": {
        "high_protein": ["Greek yogurt with berries and nuts", "Scrambled eggs with whole-wheat toast and avocado"],
        "low_carb": ["Chia pudding with almond milk and berries", "Smoked salmon with cream cheese and cucumber slices"],
        "balanced": ["Oatmeal with fruit and nuts", "Whole-wheat toast with eggs and spinach"]
    },
    "lunch": {
        "high_protein": ["Chicken breast with brown rice and roasted vegetables", "Lentil soup with whole-wheat bread"],
        "low_carb": ["Salmon with salad and avocado", "Tofu scramble with vegetables"]
    },
    "dinner": {
        "balanced": ["Salmon with roasted vegetables and quinoa", "Turkey chili with whole-wheat bread"]
    }
}
```

```
def generate_meal_plan(meal_type, intake_type):
  """
  Generates a sample meal plan based on meal type and intake type.

  Args:
      meal_type (str): The type of meal (breakfast, lunch, dinner).
      intake_type (str): The dietary preference (high-protein, low-carb, balanced).

  Returns:
      str: A string containing the generated meal plan.
  """
  if meal_type not in meal_plans or intake_type not in meal_plans[meal_type]:
      return "Sorry, we don't have meal plan options for that combination yet."
  return f"Here's a sample meal plan for {meal_type} with {intake_type} focus: \n* {meal_plans[meal_type][intake_type][0]}\n* {meal_plans[meal_type][

# Example usage
meal_plan = generate_meal_plan("lunch", "high_protein")
print(meal_plan)
```

```
from transformers import pipeline

# Load a pre-trained text generation model (replace with your desired model)
text_generator = pipeline("text-generation", model="gpt2")

# Generate text based on a prompt
prompt = "List some healthy and delicious breakfast ideas."
generated_text = text_generator(prompt, max_length=100, num_return_sequences=1)[0]["generated_text"]
print(generated_text)
```

```
import random

def augment_meal_plan(meal_plan):
  """
  Augments a meal plan by potentially replacing ingredients or adding variations.

  Args:
      meal_plan (str): The original meal plan string.

  Returns:
      str: The augmented meal plan string.
  """
  # Split the meal plan into a list of meals
  meals = meal_plan.split("\n* ")
  augmented_meals = []
  for meal in meals:
    words = meal.split()
    # Randomly replace some ingredients (replace up to 2 words)
    for i in range(2):
      if random.random() < 0.5:  # 50% chance of replacing a word
        word_index = random.randint(1, len(words)-2)  # Avoid replacing first/last word
        replacements = ["similar_ingredient_1", "similar_ingredient_2"]  # Replace with synonyms
        words[word_index] = random.choice(replacements)
    augmented_meal = " ".join(words)
    augmented_meals.append(augmented_meal)
  return f"Here are some variations of your meal plan:\n" + "\n* ".join(augmented_meals)
```

```python
def get_user_input(prompt):
    """
    Gets user input with validation and error handling.

    Args:
        prompt (str): The prompt to display to the user.

    Returns:
        str: The validated user input.
    """
    while True:
        user_input = input(prompt)
        valid_options = ["breakfast", "lunch", "dinner"]  # Example valid options
        if user_input.lower() in valid_options:
            return user_input.lower()  # Convert to lowercase for consistency
        else:
            print(f"Invalid option. Please choose from: {', '.join(valid_options)}")

# Example usage
meal_type = get_user_input("Enter the meal type (breakfast, lunch, dinner): ")
print(f"You selected: {meal_type}")
```