Artificial Intelligence Dietician

INT-404 PROJECT REPORT

By

Registration No	Name of Student	Roll No	Section
11805091	Loknath Das	56	K18FG(GROUP-2)

School of Computer Science Engineering
Lovely Professional University, Jalandhar
04-2020

TABLE OF CONTENTS

TITLE   PAGE NO.

ABSTRACT

A simple Artificial Intelligence Dietician built using Python. This project aims to take a data from user as input regarding their current physique like height, weight, and give them daily diet chart according to their BMI. After Entering age, user have too select their workout plan, this uses simple calculation and according to your workout plan diet will be suggested to you with the amount of food you should consume to remain healthy. Python function i.e. radiant has been used in this project and module tkinter is also used for GUI, the radiant function is used to select an item from the range as we made a list according to the calorie which user should consume to remain healthy. Basic BMR calculation is used to add it with exercise level as per user selection, five option has been given to the user regarding the daily activity which will be calculated with BMR (Basal Metabolic Rate) to produce a diet chart. The BMR function is used to store the amount of food and their quantity eg:-protein, fruit, vegetable, grains, etc. there is no storage base or data model for this project as it runs on simple calculation and prediction. Based on user input. If and else is used for condition mainly 4 if statement are used which have argument as calories, that means if user is male or female and their daily activity is also different then the conditions will be used as per the user input, tkinter is used to make basic gui to select things like gender, and daily activity rest all are given by the user input a button has been added in the end to submit the user data and calculate and give user their diet plan according to their input. Listbox height is set to 5 and width is set to 50, different grid is made for different value.

At last when user will put input, the interface will show the result of how much food user should consume based on their selection and show them the different food they can eat to maintain nutrient value in their body, it act as a real dietician.

Introduction

The Artificial Intelligence Dietician is a bot with artificial intelligence about human diets.

It acts as a real diet consultant similar to a real life dietician.

Dieticians are educated and well versed with food and their nutrient values.

 A dietician consults a person based on his schedule, body type, height and weight and the amount of work or activity a person does in their daily day to day life, mainly dietician consult them regarding the nutrients they should consume to cope up with body need.

The AI based system too asks the data mentioned from the user like how many hours the user works, his height, weight, age, and what amount of daily activity he does which is divided into

Four category and submit all the data, and processes it.

 It then calculates the nutrient value needed to satisfy user needs as per their physique and daily activity to keep the balance in their diet.

 Based on the rules defined and calculations, the system shows an appropriate diet to the users and asks for affirmation regarding the same in the form of input and submit.

In case the user is not satisfied, an alternate diet is provided for satisfying users' needs.

All this makes it an AI based dietician.

Objectives of Project

Objective of the project is to create an AI Dietician which works as per user input and requirement, and also works on knowledge base so it can remember the data and also make use of that knowledge available to it. A simple GUI based model is Built using python. This project aims to take a user detailed input regarding their physique and suggest them diet plan for better health and lifestyle.

An intelligent agent needs knowledge about the real nutrient values and calories for taking decisions and reasoning to act efficiently based on the user input which is given in the program in form of condition and simple mathematical calculation based model.

As a real Dietician this system needs to be accurate in terms of diet plan as the wrong plan can put user into major health issue, this is done with the help of BMR calculation and calories user need to satisfy their daily nutrient quota according to their daily day to day life activity.

At last when the input is given by the user, the interface shows the result of user and also includes how much they need to eat in order to sustain the nutrient value their body needs. And show him the exact amount of protein and all other aspect they need to consider while eating their lunch, breakfast, dinner and snack with the food they should consider eating.

METHODOLOGY SELECTED:

For the AI based dietician project, a series of steps were followed for the system to recommend a diet plan like the real life dietician would do.

Step 1: Clean and prepare the data as per the requirements

Step 2: Input details from the user: Age (in years), Weight (in kg), Height (cms.) and daily exercise level.

Step 3: Calculate Basal Metabolic Rate (BMR) using the Harris-Benedict Equations: Men BMR= 88.362 + (13.397 * weight in kg) + (4.799 * height in cm) – (5.677 * age in years) Women BMR = 447.593 + (9.247 * weight in kg) + (3.099 * height in cm) – (4.330 * age in years)

Step 4: Calculate the calorie intake using the following table:

Exercise level Daily Calories Required (Kcal/day) Little to no exercise Daily kilocalories needed = BMR x 1.2 Light exercise (1–3 days per week) Daily kilocalories needed = BMR x 1.375 Moderate exercise (3–5 days per week) Daily kilocalories needed = BMR x 1.55 Heavy exercise (6–7 days per week) Daily kilocalories needed = BMR x 1.725 Very heavy exercise (twice per day, extra heavy workouts) Daily kilocalories needed = BMR x 1.9

Step 5: Then we use radiant (0, 6) function to select items from list

Step 6: Recommend a diet plan based on the above steps. If not interested, look for an alternative plan using the defined rules by clicking on submit.
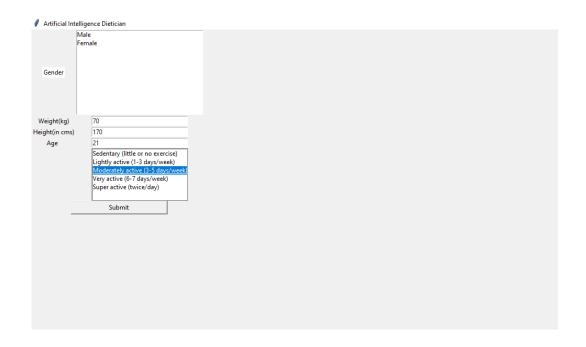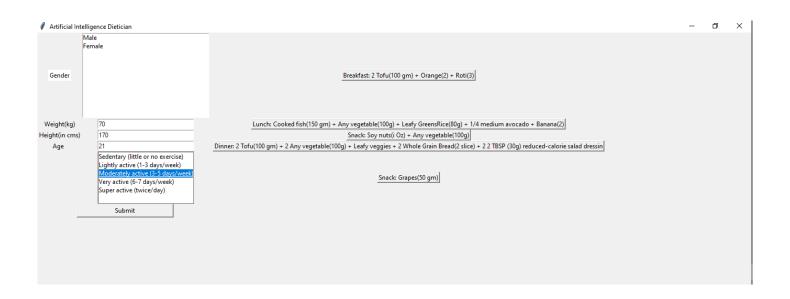
```python
from tkinter import *
from random import randint
a = Tk()
a.title('Artificial Intelligence Dietician')

def BMR():

    protein = ['EGG(2 Boiled)','Boiled Chicken(150 gm)','Cooked fish(150 gm)','1 whole egg + 4
egg whites','Tofu(100 gm)']
    fruit = ['Grapes(50 gm)','Apple(1)','Orange(2)','Banana(2)','Dried Fruits(Handfull)','Fruit
Juice(200ml)']
    vegetable = ['Any vegetable(100g)']
    grains = ['Rice(80g)','Whole Grain Bread(2 slice)','Roti(3)','Oats(200g)','2 Raagi(50g)']
    ps = ['Soy nuts(i Oz)','Low fat milk(250ml)',' cheese (125g)','Greek yogurt(125g)']
    taste_en = ['2 TSP (10 ml) olive oil','2 TBSP (30g) reduced-calorie salad dressin','1/4
medium avocado','Small handful of nuts','1/2 ounce  grated Parmesan cheese','1 TBSP (20g) jam,
jelly, honey, syrup, sugar']


    w = v3.get()
    h = v4.get()
    age = v5.get()
    act = str(Lb.get(ACTIVE))
    gender = Lb2.get(ACTIVE)

    if gender == 'Male':
        cal = float()
        cal = 88.362 + (13.397*float(w)) + (4.799*float(h)) - (5.677*float(age))
        print (cal)
    elif gender == 'Female':
        cal = float()
        cal = 447.593 + (9.247*float(w)) + (3.098*float(h)) - (4.330*float(age))


    if act == 'Sedentary (little or no exercise)':
        cal = cal*1.2

    elif act == 'Lightly active (1-3 days/week)':
        cal = cal*1.375

    elif act == 'Moderately active (3-5 days/week)':
        cal = cal*1.55

    elif act == 'Very active (6-7 days/week)':
        cal = cal*1.725

    elif act == 'Super active (twice/day)':
        cal = cal*1.9

    print (cal)


    if cal<1500:
        fin = StringVar()
        l6 = Label(a, textvariable=fin, relief=RAISED )
```

```python
            fin.set("Breakfast: "+protein[randint(0, 5)]+" + "+fruit[randint(0, 5)])
            l6.grid(row=0,column=3)


            fin2 = StringVar()
            l8 = Label(a, textvariable=fin2, relief=RAISED )
            fin2.set("Lunch: "+protein[randint(0, 5)]+" + "+vegetable[0]+" + Leafy
Greens"+grains[randint(0,4)]+" + "+taste_en[randint(0,5)])
            l8.grid(row=1,column=3)


            fin3 = StringVar()
            l9 = Label(a, textvariable=fin3, relief=RAISED )
            fin3.set("Snack: "+ps[randint(0, 4)]+" + "+vegetable[0])
            l9.grid(row=2,column=3)


            fin4 = StringVar()
            l10 = Label(a, textvariable=fin4, relief=RAISED )
            fin4.set("Dinner: "+protein[randint(0, 5)]+" + 2 "+vegetable[0]+" + Leafy
Greens"+grains[randint(0,4)]+" + "+taste_en[randint(0,5)])
            l10.grid(row=3,column=3)


            fin5 = StringVar()
            l11 = Label(a, textvariable=fin5, relief=RAISED )
            fin5.set("Snack: "+fruit[randint(0, 5)])
            l11.grid(row=4,column=3)

    elif cal<1800:
            fin = IntVar()
            l6 = Label(a, textvariable=fin, relief=RAISED )
            fin.set("Breakfast: "+protein[randint(0, 5)]+" + "+fruit[randint(0, 5)])
            l6.grid(row=0,column=3)


            fin2 = StringVar()
            l8 = Label(a, textvariable=fin2, relief=RAISED )
            fin2.set("Lunch: "+protein[randint(0, 5)]+" + "+vegetable[0]+" + Leafy
Greens"+grains[randint(0,4)]+" + "+taste_en[randint(0,5)]+" + "+fruit[randint(0, 5)])
            l8.grid(row=1,column=3)


            fin3 = StringVar()
            l9 = Label(a, textvariable=fin3, relief=RAISED )
            fin3.set("Snack: "+ps[randint(0, 4)]+" + "+vegetable[0])
            l9.grid(row=2,column=3)


            fin4 = StringVar()
            l10 = Label(a, textvariable=fin4, relief=RAISED )
            fin4.set("Dinner: 2 "+protein[randint(0, 5)]+" + "+vegetable[0]+" + Leafy
Greens"+grains[randint(0,4)]+" + "+taste_en[randint(0,5)])
            l10.grid(row=3,column=3)


            fin5 = StringVar()
            l11 = Label(a, textvariable=fin5, relief=RAISED )
            fin5.set("Snack: "+fruit[randint(0, 5)])
            l11.grid(row=4,column=3)

    elif cal<2200:
            fin = StringVar()
```

```python
        l6 = Label(a, textvariable=fin, relief=RAISED )
        fin.set("Breakfast: "+protein[randint(0, 5)]+" + "+fruit[randint(0, 5)])
        l6.grid(row=0,column=3)


        fin2 = StringVar()
        l8 = Label(a, textvariable=fin2, relief=RAISED )
        fin2.set("Lunch: "+protein[randint(0, 5)]+" + "+vegetable[0]+" + Leafy
Greens"+grains[randint(0,4)]+" + "+taste_en[randint(0,5)]+" + "+fruit[randint(0, 5)])
        l8.grid(row=1,column=3)


        fin3 = StringVar()
        l9 = Label(a, textvariable=fin3, relief=RAISED )
        fin3.set("Snack: "+ps[randint(0, 4)]+" + "+vegetable[0])
        l9.grid(row=2,column=3)


        fin4 = StringVar()
        l10 = Label(a, textvariable=fin4, relief=RAISED )
        fin4.set("Dinner: 2 "+protein[randint(0, 5)]+" + 2 "+vegetable[0]+" + Leafy
Greens"+grains[randint(0,4)]+" + "+taste_en[randint(0,5)])
        l10.grid(row=3,column=3)


        fin5 = StringVar()
        l11 = Label(a, textvariable=fin5, relief=RAISED )
        fin5.set("Snack: "+fruit[randint(0, 5)])
        l11.grid(row=4,column=3)


    elif cal>=2200:
        fin = StringVar()
        l6 = Label(a, textvariable=fin, relief=RAISED )
        fin.set("Breakfast: 2 "+protein[randint(0, 5)]+" + "+fruit[randint(0, 5)]+" +
"+grains[randint(0,4)])
        l6.grid(row=0,column=3)


        fin2 = StringVar()
        l8 = Label(a, textvariable=fin2, relief=RAISED )
        fin2.set("Lunch: "+protein[randint(0, 5)]+" + "+vegetable[0]+" + Leafy
Greens"+grains[randint(0,4)]+" + "+taste_en[randint(0,5)]+" + "+fruit[randint(0, 5)])
        l8.grid(row=1,column=3)


        fin3 = StringVar()
        l9 = Label(a, textvariable=fin3, relief=RAISED )
        fin3.set("Snack: "+ps[randint(0, 4)]+" + "+vegetable[0])
        l9.grid(row=2,column=3)


        fin4 = StringVar()
        l10 = Label(a, textvariable=fin4, relief=RAISED )
        fin4.set("Dinner: 2 "+protein[randint(0, 5)]+" + 2 "+vegetable[0]+" + Leafy veggies + 2
"+grains[randint(0,4)]+" + 2 "+taste_en[randint(0,5)])
        l10.grid(row=3,column=3)


        fin5 = StringVar()
        l11 = Label(a, textvariable=fin5, relief=RAISED )
        fin5.set("Snack: "+fruit[randint(0, 5)])
        l11.grid(row=4,column=3)
```

```python
'''
v1 = IntVar()
c1 = Checkbutton(a, text = 'Male', variable = v1)
c1.grid(row=0,column=1)
v2 = IntVar()
c2 = Checkbutton(a, text = 'Female', variable = v2)
c2.grid(row=0,column=2)
'''

l1 = Label(a, text='Weight(kg)')
l2 = Label(a, text='Height(in cms)')
l3 = Label(a, text='Age   ')
l4 = Label(a, text = 'Gender', bg = 'white')
l5 = Label(a, text = 'Activity', bg = 'white')
l7 = Label(a, text = '')


v3=StringVar()
v4=StringVar()
v5=StringVar()


e3 = Entry(a, textvariable=v3, width=30)
e4 = Entry(a, textvariable=v4, width=30)
e5 = Entry(a, textvariable=v5, width=30)

Lb = Listbox(a, height=6, width=30)
Lb.insert(1, 'Sedentary (little or no exercise)')
Lb.insert(2, 'Lightly active (1-3 days/week)')
Lb.insert(3, 'Moderately active (3-5 days/week)')
Lb.insert(4, 'Very active (6-7 days/week)')
Lb.insert(5, 'Super active (twice/day)')

Lb2 = Listbox(a, height=10, width=40)
Lb2.insert(1, 'Male')
Lb2.insert(2, 'Female')

var = Lb.get(ACTIVE)
print (var)

l5 = Label(a, text = '')
l5.grid(row=5,column=0)

b1 = Button(a, text = 'Submit', width=25, command = BMR)

l1.grid(row=1,column=0)
l2.grid(row=2,column=0)
l3.grid(row=3,column=0)
l4.grid(row=0,column=0)
l5.grid(row=4,column=0)
l7.grid(row=0,column=2)
e3.grid(row=1, column=1)
e4.grid(row=2, column=1)
e5.grid(row=3, column=1)
Lb.grid(row=4, column = 1)
Lb2.grid(row=0, column = 1)
b1.grid(row=6,columns=3)


a.mainloop()
```

Artificial Intelligence Dietician

Gender

Male
Female

Weight(kg) 70
Height(in cms) 170
Age 21

Sedentary (little or no exercise)
Lightly active (1-3 days/week)
Moderately active (3-5 days/week)
Very active (6-7 days/week)
Super active (twice/day)

Submit

Artificial Intelligence Dietician

Gender

Male
Female

Weight(kg) 70
Height(in cms) 170
Age 21

Sedentary (little or no exercise)
Lightly active (1-3 days/week)
Moderately active (3-5 days/week)
Very active (6-7 days/week)
Super active (twice/day)

Submit

Breakfast: 2 Tofu(100 gm) + Orange(2) + Roti(3)

Lunch: Cooked fish(150 gm) + Any vegetable(100g) + Leafy GreensRice(80g) + 1/4 medium avocado + Banana(2)

Snack: Soy nuts(i Oz) + Any vegetable(100g)

Dinner: 2 Tofu(100 gm) + 2 Any vegetable(100g) + Leafy veggies + 2 Whole Grain Bread(2 slice) + 2 2 TBSP (30g) reduced-calorie salad dressin

Snack: Grapes(50 gm)

CONCLUSION

Artificial Intelligence Dietician" allow the user to know about his/her actual diet information i.e. how much user had calories in their body on this basis system displays food suggestions. Building an AI based Dietician system is essential. Any customer or user nowadays demands and deserves immediate things regarding their health.

As per large number of population it is impossible for dietician to suggest everyone what should they eat to keep themselves in good shape.

Therefore having a smart and accurate AI based system would make this task easy and hassle free. Also it would be easy for billions of people to stay healthy without consulting dietician every time for diet plan.

So ultimately, having a fast and accurate AI based system would be beneficial to everyone.

All of this issue can be remove and improved using AI Dietician.

FUTURE SCOPE

 The future scope in this Project are going to be the Basic GUI with python and proper workout schedule and routine will be provided to the user for a better fit life. Less use of data base and conditioning algorithms and more use of Artificial Intelligence.

1. It will help Dietician to ease up their work load
2. Reduce of number of mistakes
3. Reduce the cost required to consult a real life Dietician
4. Help masses to stay fit and healthy
5. Reduce effort of user to go through a list of checklist regarding health.

In today's busy and competitive world, fast and accurate health system or dietician is important to keep the world population healthy.

GITHUB- https://github.com/cosmoscode183/AI_DIETICIAN
YOUTUBE- https://youtu.be/8P72l8q-rMs