VAMSTAR

Python BMI Calculator Coding Challenge V6

Problem Statement

Given the following JSON data

```
[{"Gender": "Male", "HeightCm": 171, "WeightKg": 96 },
 { "Gender": "Male", "HeightCm": 161, "WeightKg": 85 },
 { "Gender": "Male", "HeightCm": 180, "WeightKg": 77 },
 { "Gender": "Female", "HeightCm": 166, "WeightKg": 62},
 {"Gender": "Female", "HeightCm": 150, "WeightKg": 70},
 {"Gender": "Female", "HeightCm": 167, "WeightKg": 82}]
```

as the input with weight and height parameters of a person, we have to perform the following:

1) Calculate the BMI (Body Mass Index) using *Formula 1*, BMI Category and Health risk *from Table 1* of the person and add them as 3 new columns
2) Count the total number of overweight people using ranges in the column BMI Category of *Table 1*, check this is consistent programmatically and add any other observations in the documentation

3) Create build, tests to make sure the code is working as expected and this can be added to an automation build / testing / deployment pipeline

Formula 1 - BMI

$BMI(kg/m^2) = mass(kg) / height(m)^2$

The BMI (Body Mass Index) in (kg/m^2) is equal to the weight in kilograms (kg) divided by your height in meters squared $(m)^2$. For example, if you are `175cm (1.75m)` in height and `75kg` in weight, you can calculate your BMI as follows: `75kg / (1.75m²) = 24.49kg/m²`

Table 1 - BMI Category and the Health Risk.

BMI Category	BMI Range (kg/m2)	Health risk
Underweight	18.4 and below	Malnutrition risk
Normal weight	18.5 - 24.9	Low risk
Overweight	25 - 29.9	Enhanced risk
Moderately obese	30 - 34.9	Medium risk
Severely obese	35 - 39.9	High risk
Very severely obese	40 and above	Very high risk

Challenge

1) Write a **solid production-grade Python3** Program to solve this problem, imagine this will be used in-product for 1 Lac patients. We are only interested in a standalone backend application, we are NOT expecting a UI, webpage, frontend, Mobile App, microsite etc. We want to see what optimal solution you come up with to scale for larger JSON data and perform calculations quickly and write the output efficiently. Feel free to explore and use the standard Python libraries or any open source Python modules
2) Automate the **setup, build, testing, package** and run using your favourite tools

3) Check in the documentation, configuration, code and tests into github and please send us the link as `https://www.github.com/<owner>/code-<date>-<your fullname>` and do NOT use Vamstar in title or description

Evaluation Criterion

We will be evaluating your project with the following:

- 25% **Working code** and Python Programming Knowledge and clean code, reuse
- 25% Problem Analysis and **Solution Approach**

- 25% **Build and Testing** Automation Approach
- 25% **Originality**, we deduct marks or reject any projects with directly copied or plagiarised code - we have an NLP process to detect this

For the coding challenge submissions, please do NOT use Vamstar or BMI in the public repository name or description instead follow the URL pattern as follows:

`https://www.github.com/<owner>/code<date>-<your fullname>` e.g. for me it could be `https://www.github.com/richard/code-20200917-richardfreeman`

Please get back to us if you have issues or doubts, you have upto 1 week to complete this task.